

СТАБИЛИЗАЦИЯ МАТЕМАТИЧЕСКОГО МАЯТНИКА С ОСНОВАНИЕМ НА КОЛЕСЕ С ПОМОЩЬЮ НЕЧЕТКОГО АЛГОРИТМА УПРАВЛЕНИЯ

В.И. Ухоботов, В.С. Величко

Излагается задача стабилизации верхнего положения равновесия перевернутого однозвенного маятника на колесе. Маятник расположен на горизонтальной плоскости и на него действуют внешние силы. Исследуется механика движения маятника на колесе. Строится математическая модель. Производится вывод уравнений движения системы. Приводятся используемые методы обработки нечетких правил. Рассматривается задача о компьютерном моделировании механизма с помощью пакета программ моделирования физических законов. Описывается итоговая программа, которая производит построение визуализации механизма, позволяет изменять начальные параметры системы, предоставляет возможность интерактивно создавать и изменять внешние возмущения. Исследуется задача управления моделью с помощью системы нечетких правил. Приводится использованная система правил, позволяющая стабилизировать верхнее неустойчивое положение маятника. Производится анализ полученной системы управления модели. Рассматривается особый случай поведения механизма, в котором система схожа с моделью руки человека, вертикально поднятой вверх.

Ключевые слова: перевернутый маятник, нечеткая логика, равновесие.

Введение

Исследованию движения многозвенных механизмов посвящено большое количество литературы. Весьма актуальна задача компьютерного управления поведением механических объектов. Если рассматривать эти объекты и управляющий ими компьютер как единое целое, то эту задачу можно сформулировать как попытку создания объекта, способного самостоятельно управлять своими действиями.

Настоящая статья посвящена задаче управления перевернутым маятником на колесе: требуется привести маятник в верхнее неустойчивое положение и стабилизировать в этом положении для произвольных начальных условий при ограниченном управлении на базе нечеткой логики [1–3]. За основу работы была взята статья [4], в которой рассматривается механика движения маятника на колесе. Исходя из уравнений движения были получены нечеткие правила для управления системой.

1. Уравнения движения

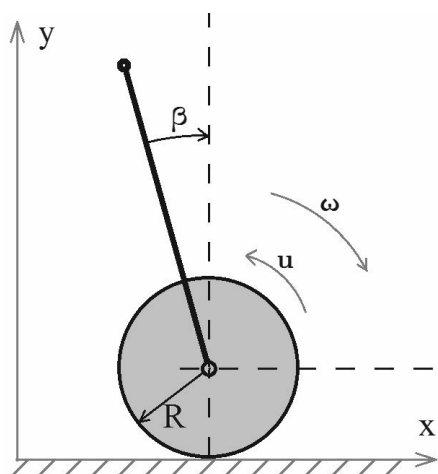


Рис. 1. Маятник на колесе

Рассматриваемая конструкция маятника на колесе изображен на рис. 1. Она состоит из колеса, свободно движущегося по прямой, и маятника в виде стержня. Маятник относительно колеса может приводиться во вращение управляемым электродвигателем. Управление движением маятника осуществляется за счет изменения направления и скорости вращения приводного двигателя под действием управляющего напряжения питания в цепи якоря. Это управление задается в соответствии с поступающими от компьютера командами и ограничено по модулю. Управляющий момент, приложенный между колесом и звеном маятника, не зависит от положения колеса и его скорости и ограничен по абсолютной величине. При этом из математической модели системы можно выделить уравнения, описывающие только движение маятника. В статье [4] построено в виде обратной связи управление маятником, обеспечивающее глобальную стабилизацию его верхнего неустойчивого положения равновесия.

Уравнение движения маятника на колесе имеет следующий вид [4]:

$$\beta' = \omega, \quad (1 - d^2 \cos^2 \beta) \omega' + d^2 \omega^2 \sin \beta \cos \beta - \sin \beta = (1 + e^2 \cos \beta) u. \quad (1)$$

Здесь $d^2 = \frac{m_1 (b_1 / r_1)^2}{m_0 [1 + (r_0 / R)^2] + m_1}$; $e^2 = d^2 (r_1 / R)$; m_0 – масса колеса; m_1 – масса маятника; M – мо-

мент; β – угол отклонения маятника; ω – угловое ускорение. Наличие члена $e^2 \cos \beta$ в правой части второго уравнения (1) связано с тем, что момент M прикладывается не только к маятнику, но и к колесу, вызывая ускорение точки O , которое, в свою очередь, влияет на движение маятника. В диапазоне $-\pi/2 < \beta < \pi/2$ это ускорение направлено в одну сторону, а в диапазоне $\pi/2 < \beta < 3\pi/2$ при этом же значении u – в другую сторону. Следовательно, член $e^2 \cos \beta$ при $\pi/2 < \beta < 3\pi/2$ усиливает влияние управления, а при $\pi/2 < \beta < 3\pi/2$ ослабляет.

Интеграл энергии системы имеет вид $E = [(1 - d^2 \cos^2 \beta) \omega^2 + \cos \beta] / 2 = \text{const}$. Полная производная энергии по времени

$$E' = (1 - e^2 \cos^2 \beta) \omega u. \quad (2)$$

Производная (2) максимальна при управлении [4]

$$u = u_0 \text{sign}[(1 + e^2 \cos^2 \beta) \omega]. \quad (3)$$

Если $e^2 < 1$, то знак управляющего воздействия u определяется только знаком скорости ω . Если же $e^2 > 1$, то управление u меняет знак «около» положения $\beta = \pi$, независимо от знака скорости ω .

При управлении (3) энергия E монотонно возрастает и в некоторый момент времени становится равной единице. При $u = 0$ после этого момента система будет стремиться к $\beta = 0, 2\pi$; $\omega = 0$.

2. Физическое моделирование

2.1. Среда моделирования

Графическая и физическая модели построены при помощи библиотеки Vox2D 2.0 [5]. Это библиотека физического моделирования и визуализации с реализациями на наиболее распространенных языках программирования с открытым исходным кодом. С ее помощью программирование двумерной графики становится более простым и быстрым. Эта библиотека промышленного качества, предназначенная для использования в интерактивных симуляторах, а также симуляторах реального времени, хорошо подходит для моделирования шарнирно связанных твердых тел. Например, она хорошо подходит для транспортных средств (у которых колеса соединены с подвеской), существ с ногами и движущихся объектов в виртуальном пространстве. Она быстра, гибка и проста. Встроенная система обнаружения столкновений позволяет моделировать взаимодействие тел при их столкновении. В Vox2D также реализована симуляция сухого трения между телами.

При выводе уравнения движения (1) в работе [4] не допускается проскальзывание колеса относительно поверхности. При создании среды для физического моделирования допускается такое проскальзывание. Это сделано для большей правдоподобности физического процесса и увеличения нечеткости в характере модели.

2.2. Создание компьютерной модели

В качестве среды программирования использована Embarcadero Delphi XE2. Для Delphi существует реализация библиотеки Vox2D.

В целях упрощения анализа поведения модель была взята максимально простой. Однако, чтобы сохранялись все основные свойства ее поведения, модель должна оставаться похожей на реальную.

Для каждого элемента модели создается объект с заданными габаритами, массой, степенями свободы и т. п. На каждом шаге моделирования для каждой пары объектов обрабатываются столкновения и силы взаимодействия.

3. Обработка нечетких правил

При построении нечеткой системы управления для каждой физической переменной были определены соответствующие лингвистические переменные с выбранными именами. Для каждой переменной были взяты пять термов: *verynegative*, *negative*, *zero*, *positive*, *verypositive* (очень отрицательно, отрицательно, ноль, положительно, очень положительно соответственно).

Графический вид этих переменных может отличаться лишь смещением узлов в зависимости от параметров конкретного маятника (рис. 2).

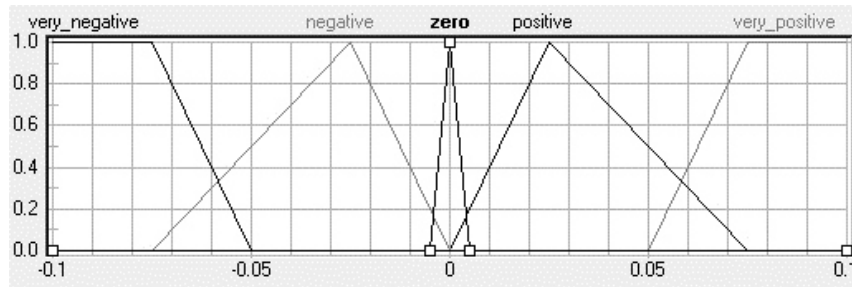


Рис. 2. Общий графический вид термов переменных

При этом количество правил получается равным $5^2 = 25$ (количество всевозможных комбинаций всех термов всех входных переменных). Нечеткие правила и функции принадлежности составлены в системе FuzzyTECH 5.54 [6].

При фаззификации используется стандартный метод фаззификации (Compute MBF). Он предполагает использование функций принадлежности стандартного типа – треугольных, трапециевидных и кусочно-линейных кривых.

Для агрегирования подусловий правил нечетких продукций было применено правило минимума $b_k'' = \min_{i \in \{1, 2, \dots, m\}} \{\mu(a_i)\}$.

Для композиции заключений правил был использован стандартный (Standart) вариант задания весовых коэффициентов правил, при котором значение весового коэффициента для каждого правила предполагается неизменным и равным 1.

Для аккумуляции заключений правил был выбран метод граничной суммы (Bsum), при котором результат нечеткого вывода в блоке правил определяется как объединение нечетких множеств.

Для дефаззификации выходной переменной u используется стандартный метод (Center_off_Maximum или сокращенно CoM). В программе fuzzyTECH метод дефаззификации CoM работает аналогично методу центра тяжести CoG, определяемому по формуле

$$y = \frac{\int_{\min}^{\max} x \cdot \mu(x) dx}{\int_{\min}^{\max} \mu(x) dx}.$$

Здесь y – результат дефаззификации; x – переменная, соответствующая выходной лингвистической переменной; $\mu(x)$ – функция принадлежности нечеткого множества, соответствующего выходной переменной после этапа аккумуляции; \min и \max – левая и правая точки интервала носителя нечеткого множества рассматриваемой выходной переменной.

Например, если маятник отклонен на небольшой положительный угол и при этом скорость его поворота сильно отрицательна, то необходимо «притормозить» маятник управлением $u > 0$ ($u = +$), чтобы он не проскочил верхнее вертикальное положение. Примеры нечетких правил:

IF $\beta = positive$ AND $\omega = verynegative$ THEN $u = positive$;

IF $\beta = verypositive$ AND $\omega = verypositive$ THEN $u = verynegative$.

Нечеткие правила можно представить в виде таблицы, в которой положительные и очень положительные значения переменных обозначены как + и ++, а отрицательные и очень отрицательные – и -- соответственно. Нулем обозначим значения близкие к нулю.

Нечеткие правила

$\beta \backslash \omega$	--	-	0	+	++
--	++	++	++	+	0
-	++	++	+	0	-
0	+	+	0	-	-
+	+	0	-	--	--
++	0	-	--	--	--

Те же правила можно представить в виде трехмерной поверхности (рис. 3), где по горизонтальным осям расположены входные переменные β и ω , а на вертикальной оси – управление u .

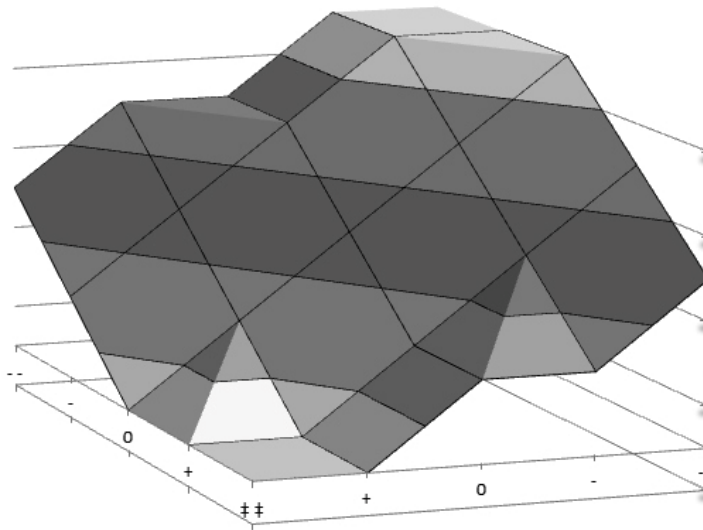


Рис. 3. Представление термов в виде трехмерной поверхности

При программировании используем модуль обработки в реальном времени FTRUN32.DLL из пакета FuzzyTECH 5.54.

Использование модуля обработки в коде программы на Delphi:
ftrOpen (hinstance, 'BIKE.ftr', hftrTra_c).

Передавая входные параметры системы управления, получаем выходные на каждом шаге управления (каждые 1–2 мс):

```
ftrSetShellValue (hftrTrafc, 0, beta);  
ftrSetShellValue (hftrTraffc, 1, omega);  
ftrGetShellValue (hftrTraffc, 2, u, NoHitFlag).
```

4. Итоговая программа

При запуске программы на черном фоне виден сам механизм на зеленой плоскости (рис. 4). В правом верхнем углу располагается панель, с полями для изменения направления и силы тяжести, кнопки сброса системы в исходное состояние, остановки системы и пошагового движения. На панели также отображается текущее значение управления.

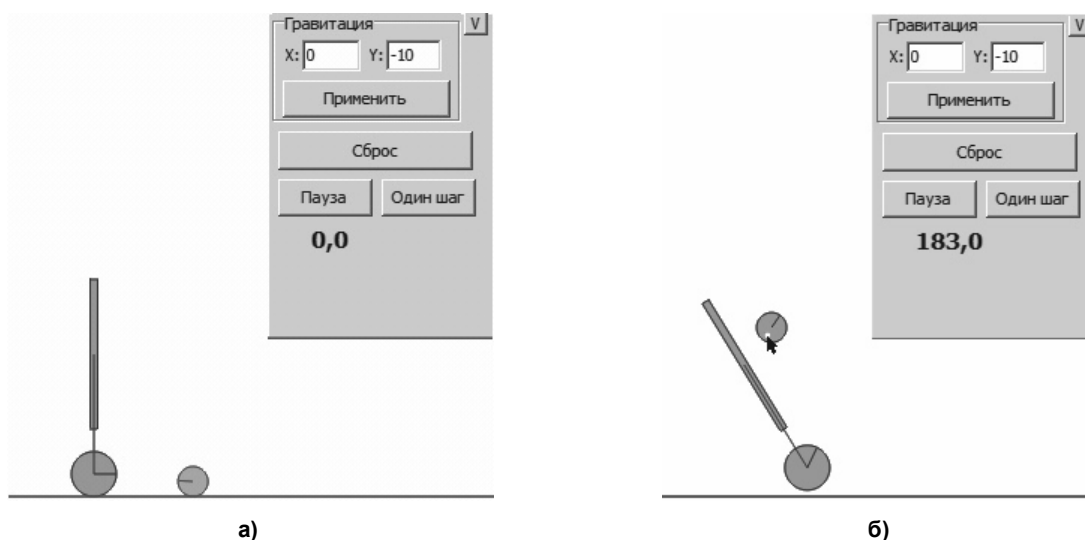


Рис. 4. Вид окна программы: а – маятник в спокойном (устойчивом) состоянии; б – маятник выведен из состояния равновесия

С помощью левой кнопки мыши, можно «хватать» маятник за звено и за колесо для внесения помех в систему и вывода системы из равновесия.

На поверхности кроме маятника на колесе также расположено круглое тело, «схватив» которое курсором мыши и ударяя по маятнику, можно вызывать изменения состояния системы (рис. 4, а).

Для удобства просмотра масштабом отображения можно управлять с помощью колеса мыши. Правой кнопкой мыши, хватая за фон, можно перемещать отображаемую область. Если же маятник вышел за область видимости и его поиск затруднителен, то в любой момент времени можно нажать кнопку сброса, приводящую систему в исходное состояние.

Для выхода из программы необходимо нажать *Escape*.

5. Анализ полученной системы управления модели

Без вмешательства пользователя система находится в состоянии покоя, так как маятник находится в верхнем неустойчивом положении равновесия.

При внесении помех система управления приводит маятник в вертикальное положение.

При попытке задать силу тяжести близкую к нулю, система способна привести маятник в вертикальное (относительно поверхности) положение.

Особого внимания заслуживает случай, когда курсором мыши можно подвешивать маятник за колесо, приподнимая над поверхностью и фиксируя точку подвеса (рис. 5, а). Управление справляется с поставленной задачей, несмотря на то, что система значительно изменяется и систему можно представить в виде, представленном на рис. 5, б).

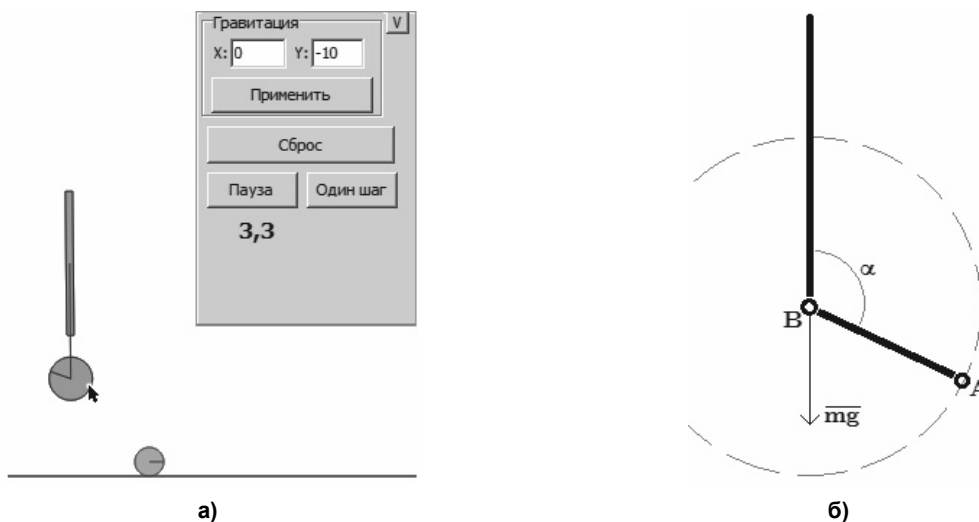


Рис. 5. Иное представление системы: а – вид окна программы; б – шарнирно соединенные звенья

Система в этом случае будет состоять из двух стержневых звеньев, шарнирно соединенных в точке В. Один из стержней шарнирно соединен в жестко фиксированной точке А. Такая система может являться моделью руки человека, вертикально поднятой вверх. Можно рассматривать задачу вертикального удержания «предплечья».

Заключение

В работе построена на основании нечеткой логики система управления маятника на колесе. Она стабилизирует верхнее неустойчивое положение маятника, при наличии внешних возмущений. Эта система управления обеспечивает равновесие маятника в широком диапазоне его параметров.

Литература

1. Заде, Л.А. Понятие лингвистической переменной и его применение к принятию приближенных решений / Л.А. Заде. – М.: Мир, 1976. – 161 с.
2. Ухоботов, В.И. Избранные главы теории нечетких множеств : учеб. пособие / В.И. Ухоботов. – Челябинск: Изд-во ЧелГУ, 2011. – 245 с.
3. Ухоботов В.И. Моделирование заданного движения пятизвенного механизма / В.С. Величко, В.И. Ухоботов. – Вестник ЮУрГУ. Сер. «Вычислительная математика и информатика», 2013. – Т. 2, № 3. – С. 104–110.

4. Мартыненко, Ю.Г. Маятник на подвижном основании / Ю.Г. Мартыненко, А.М. Формальский // Доклады Академии наук, 2011. – Т. 439, № 6. – С. 746–751.
5. Catto, E. Box2D v2.0.2 Руководство / Erin Catto. – <http://box2d.ru/page/manual/>.
6. Леоненко, А.В. Нечеткое моделирование в среде MATLAB и fuzzyTECH / А.В. Леоненко. – СПб.: БХВ-Петербург, 2005. – 736 с.

Ухоботов Виктор Иванович, д-р физ.-мат. наук, профессор, заведующий кафедрой теории управления и оптимизации, Челябинский государственный университет (г. Челябинск); ukh@csu.ru.

Величко Владислав Станиславович, аспирант, математик УНЛ методов оптимизации и моделирования игровых ситуаций кафедры теории управления и оптимизации, Челябинский государственный университет (г. Челябинск); 124816@list.ru.

Поступила в редакцию 20 января 2014 г.

**Bulletin of the South Ural State University
Series “Computer Technologies, Automatic Control, Radio Electronics”
2014, vol. 14, no. 2, pp. 18–23**

STABILIZATION OF A MATHEMATICAL PENDULUM WITH BASE ON WHEEL WITH FUZZY CONTROL ALGORITHM

V.I. Ukhobotov, Chelyabinsk State University, Chelyabinsk, Russian Federation, ukh@csu.ru,
V.S. Velichko, Chelyabinsk State University, Chelyabinsk, Russian Federation, 124816@list.ru

The article describes the problem of stabilizing upper equilibrium for one section inverted pendulum on the wheel. The pendulum is located on the horizontal plane, the external forces act on it. The mechanical motion of the pendulum on the wheel is investigated. The mathematical model is built. Derivation of the equations of motion of the system is produced. The paper describes the used methods for the processing of fuzzy rules. The research considers the problem of computer simulation of the mechanism using a software package for modeling physics. The final program is described, that visualizes the mechanism, allows changing the input system parameters, gives an opportunity to create and vary external perturbations. The problem of control model with a system of fuzzy rules is presented. The used system of rules is shown; it allows stabilizing the upper equilibrium of pendulum. The article considers the analysis of the received system for controlling the model. The special case is studied: system is similar to the model of the raised up human hand.

Keywords: inverted pendulum, fuzzy logic, equilibrium.

References

1. Zadeh L.A. *Ponyatie lingvisticheskoy peremennoy i ego primeneniye k prinyatiyu priblizhennykh resheniy* [The Concept of a Linguistic Variable and its Application to Approximate Reasoning]. New York, American Elsevier Publishing Company, 1973. 161 p.
2. Ukhobotov V.I. *Izbrannie glavi teorii nechetkikh mnozhestv* [Selected Chapters of the Theory of Fuzzy Sets]. Chelyabinsk, ChelGU Publ., 2011. 245 p.
3. Ukhobotov V.I. [Simulation of the Given Five-section Mechanism]. *Bulletin of the South-Ural State University. Ser. Computational Mathematics and Informatics*, 2013, vol. 2, no. 3, pp. 104–110. (in Russ.)
4. Martynenko Y.G. [Pendulum on a Movable Base]. *Reports of the Academy of Sciences*, 2011, vol. 439, no. 6, pp. 746–751. (in Russ.)
5. Catto E. Box2Dv2.0.2 User Manual. Available at: <http://box2d.org/manual.html> (accessed 20 December 2013).
6. Leonenko A.V. *Nechetkoe modelirovaniye v srede MATLAB i FuzzyTECH* [Fuzzy Simulation in MATLAB and FuzzyTECH Environment]. St. Petersburg, BKhV-Peterburg Publ., 2005. 736 p.

Received 20 January 2014