

УДК 004.85

Применение генетических алгоритмов для генерации функций, описывающих движение, на примере шага вперед человекоподобного робота

Ю. К. Чеботарева

**Санкт-Петербургский государственный университет информационных технологий,
механики и оптики**
Yulia.Chebotareva@gmail.com

Создание человекоподобных роботов – одно из наиболее актуальных и перспективных направлений науки. Роботы такого типа могут эффективно выполнять различные задачи в мире, приспособленном для человека.

Международный проект *RoboCup* создан с целью привлечения внимания ученых и разработчиков к проблемам искусственного интеллекта, робототехники и смежных областей [1]. Движение *RoboCup* спонсирует онлайн-соревнование по программированию роботов-футболистов *RobotStadium* [2]. Участникам соревнования предлагается с помощью симулятора *Webots* написать управляющие программы для роботов *Nao* французской компании *Aldebaran Robotics*. Готовые программы можно загружать на сайт, где периодически проводятся соревнования между ними.

Основой системы, управляющей движением робота, является набор базовых действий: шаг вперед, шаг назад, поворот влево, вправо, удар по мячу. В то же время задать эти действия весьма непросто. Действие состоит из некоторого числа фаз (в примерах описаний движений, поставляемых вместе с симулятором, от 10 до 100 и более), каждая из которых определяет значения углов сервомоторов робота. Подобрать требуемые значения вручную практически невозможно. Методы, основанные на физике и биомеханике движений, требуют специфических знаний и сложны для реализации.

В данной работе для генерации движений роботов автор предлагает использовать генетические алгоритмы. При этом в качестве примера применения разработанного метода рассматривается задача выполнения роботом шага вперед.

Модель робота *Nao*

Модель *Nao*, используемая в симуляторе *Webots*, оснащена 22 сервомоторами. Их расположение показано на рис. 1 (сервомоторы *RWristYaw*, *LWristYaw*, *RHand* и *LHand* в симуляторе не используются). Для анализа информации об окружающей среде робот оснащен видеокамерой, акселерометром, гироскопом, датчиками давления на ногах, датчиками

соприкосновения с полом на ногах, ультразвуковыми датчиками для определения расстояния до ближайшего препятствия, а также эмиттером и ресивером.

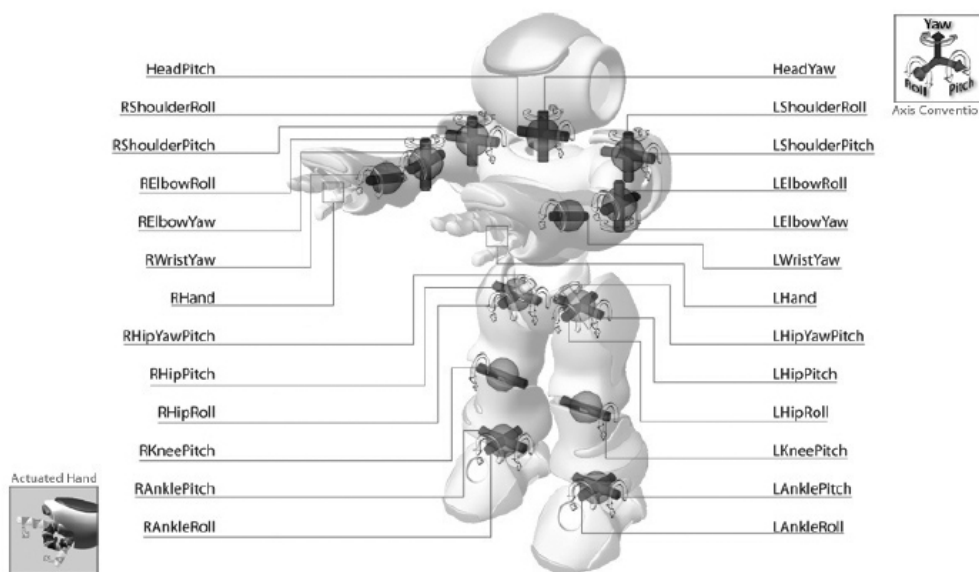


Рис. 1. Расположение сервомоторов робота

Постановка задачи

В данной работе автор ставит своей целью разработку метода генерации движения вперед для модели робота *Nao* с учетом следующих ограничений:

1. Метод должен базироваться только на информации предоставляемой средой симуляции. Это ограничение имеет важное практическое значение. Часто исследователи не могут позволить себе проведение экспериментов на настоящих роботах, так как они весьма дорогостоящи. Поэтому логично проводить эксперименты с виртуальной моделью робота, а затем уже адаптировать полученные движения для реального робота.

2. Во время проигрывания полученного движения вперед, робот должен максимально сместиться вперед, минимально отклониться и/или повернуться в сторону.

3. Для упрощения задачи будем считать, что только следующие 10 сервомоторов участвуют в выполнении шага: *RHipPitch*, *RHipRoll*, *RKneePitch*, *RAnklePitch*, *RAnkleRoll*, *LHipPitch*, *LHipRoll*, *LKneePitch*, *LAnklePitch*, *LAnkleRoll* (все они управляют ногами робота).

В качестве основы для предлагаемого метода генерации движения вперед автор выбрал генетические алгоритмы [3, 4]. При этом был изучен вопрос выбора параметров генетического алгоритма и зависимость между этими параметрами и качеством получаемых движений.

Вместе с симулятором *Webots* распространяется набор примеров движений, среди которых есть и шаг вперед. С ним и будет проведено сравнение полученных при написании работы движений.

Основы предлагаемого метода

Будем искать траектории углов сервомоторов робота в виде синусоид:

$$\alpha_i(t) = A_i \sin(\omega t) + \alpha_{i_0},$$

$\alpha_i(t)$ – угол i -го сервомотора в момент времени t ; A_i – амплитуда i -го сервомотора; ω – частота; α_{i_0} – начальное положение i -го сервомотора.

В работах [5, 6] показано, что подход, основанный на использовании синусоид, позволяет получать неплохие результаты для аналогичных роботов. Таким образом, решение поставленной задачи сводится к подбору значений параметров ω , A_i и α_{i_0} для всех участвующих в движении сервомоторов. Для того чтобы сузить пространство поиска введем дополнительные связи между искомыми значениями.

Рассмотрим сначала выполнение движения в дорсовентральной плоскости и заметим, что во время движения вес тела переносится на опорную ногу (рис. 2).

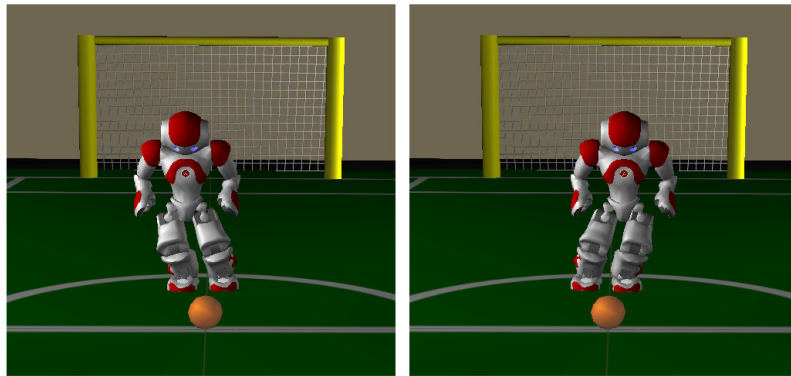


Рис. 2. Перенос веса тела с одной ноги на другую

Подобное движение может быть промоделировано изменением положений сервомоторов *HipRoll* и *AnkleRoll* в соответствии с уравнениями (рис. 3).

$$\alpha_{HipRoll}(t) = A_{HipRoll} \sin(\omega t)$$

$$\alpha_{AnkleRoll}(t) = -A_{AnkleRoll} \sin(\omega t)$$

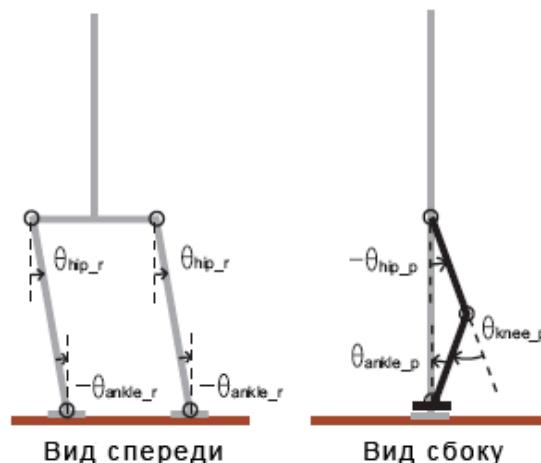


Рис. 3. Положения сервомоторов во время шага вперед

Далее рассмотрим движение вперед в латеральной (боковой) плоскости. В этой плоскости движение осуществляется за счет сервомоторов *HipPitch*, *KneePitch* и *AnklePitch*. В работе [5] для них используются выражения:

$$\begin{aligned}\alpha_{HipPitch}(t) &= A_{Pitch} \sin(\omega t) + \alpha_{HipPitch0} \\ \alpha_{KneePitch}(t) &= -2A_{Pitch} \sin(\omega t) + \alpha_{KneePitch0} \\ \alpha_{AnklePitch}(t) &= A_{Pitch} \sin(\omega t) + \alpha_{AnklePitch0}\end{aligned}$$

В ходе экспериментов было установлено, что такие выражения подходят для использования с описанной моделью робота, но коэффициенты были уточнены. Таким образом, уравнение для *AnklePitch* приобрело вид:

$$\alpha_{AnklePitch}(t) = 0.5A_{Pitch} \sin(\omega t) + \alpha_{AnklePitch0}$$

Кроме того, было замечено, что коэффициент при A_{pitch} в уравнении для сервомоторов *AnklePitch* слишком велик. Было произведено несколько попыток подобрать его вручную, и оказалось, что для разных наборов A_{pitch} , $A_{HipRoll}$ и $A_{AnkleRoll}$ оптимальными являются различные его значения из диапазона [1, 2], поэтому он был добавлен к параметрам оптимизации:

$$\alpha_{KneePitch}(t) = -kA_{Pitch} \sin(\omega t) + \alpha_{KneePitch0}, k \in [1, 2]$$

Для того чтобы обеспечить симметричность движения, «левые» и «правые» сервомоторы, двигающиеся в латеральной плоскости, принимают значения со сдвигом по фазе равной половине периода.

Несмотря на то, что уже с использованием таких простых выражений для положений сервомоторов, удастся получить неплохие результаты, в них были внесены некоторые изменения, полученные эвристически: было замечено, что при использовании одних только синусоид, робот часто оказывается неспособен полностью оторвать ногу от пола, так как он начинает это делать еще до того, как вес тела будет перенесен на другую ногу. В результате робот не шагает в традиционном понимании этого слова, а скорее скользит, что отрицательно сказывается на скорости движения (иногда робот и вовсе падает).

Для того чтобы этого избежать, положения сервомоторов *HipPitch* и *AnklePitch* для временных отрезков вида $\left[nT + \phi, T(n + \frac{1}{4}) + \phi \right]$, где $T = \frac{2\pi}{\omega}$, ϕ – сдвиг по фазе для соответствующей ноги (либо 0, либо $\frac{T}{2}$) принимаются равными $\alpha_{HipPitch0}$ и $\alpha_{AnklePitch0}$ соответственно. Графики соответствующих зависимостей показаны на рис. 4.

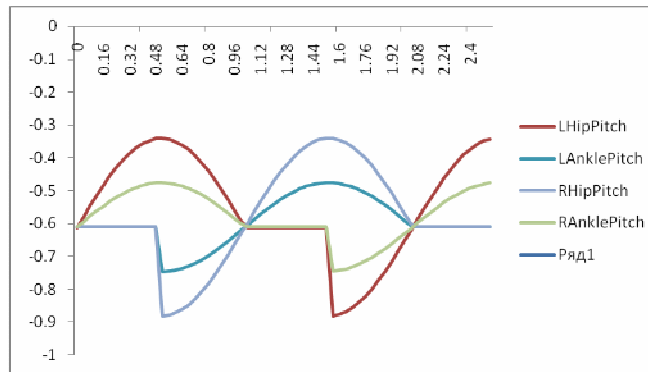


Рис. 4. Графики зависимостей положений сервомоторов *LHipPitch*, *RHipPitch*, *LAnklePitch* и *RAnklePitch* от времени

Положения углов сервомоторов в начальный момент времени были определены вручную. Таким образом, для полного задания движения требуется пять параметров: A_{Pitch} , $A_{AnkleRoll}$, $A_{HipRoll}$, k и ω . Они и будут оптимизироваться средствами генетического программирования.

Описание генетического алгоритма

Для работы генетического алгоритма необходимо реализовать кодирование особей (движений робота) в хромосомы (числовые наборы), задать генетические операторы над хромосомами выбранного вида и алгоритм вычисления функции-приспособленности.

В качестве реализации генетических алгоритмов в работе было использовано инструментальное средство, описанное в работе [7].

Структура хромосомы

Роль хромосомы в данной работе выполняет вектор из пяти вещественных чисел, соответствующих значениям A_{Pitch} , $A_{AnkleRoll}$, $A_{HipRoll}$, k и ω . Диапазон допустимых значений для параметров оптимизации приведен в табл. 1.

Таблица 1. Параметры начального движения и ограничения на параметры оптимизации

Параметр оптимизации	Минимальное значение	Максимальное значение	Начальное значение
A_{hip_roll}	0	0.3	0.1
A_{ankle_roll}	0	0.3	0.1
A_{hip_pitch}	0	0.4	0.15
k	1	2	1
T	1	6	1

Создание начального поколения

Генетические алгоритмы являются оптимизационными методами, и поэтому возникает необходимость в выборе стартовой точки – начального движения, которое будет оптимизироваться.

В данной работе параметры начального движения были подобраны вручную. Они приведены в табл. 1.

Начальное поколение заполняется N особями, каждая из которых является результатом применения мутации к начальной особи, которая оптимизируется.

Генетические операторы

Оператор скрещивания получает на вход две особи из предыдущего поколения, которые выбираются по принципу «рулетки», и возвращает также две особи. В работе использовались два оператора скрещивания: арифметический оператор скрещивания и смешанный оператор скрещивания [8].

Арифметический оператор скрещивания. Пусть c_1, c_2 – хромосомы выбранные оператором селекции для проведения скрещивания. Создаются два потомка: $H_1 = (h_1^1, \dots, h_n^1)$ и $H_2 = (h_1^2, \dots, h_n^2)$, такие что $h_i^1 = \eta c_i^1 + (1 - \eta)c_i^2$, $h_i^2 = \eta c_i^2 + (1 - \eta)c_i^1$, $i = \overline{1, n}$, η – случайное число из интервала $[0, 1]$.

Смешанный кроссовер (BLX-alpha crossover). Генерируются два потомка $H_i = (h_1^i, \dots, h_n^i), i = \overline{1, 2}$, где $[c_{\min} - \alpha I, c_{\max} + \alpha I], c_{\min} = \min(c_k^1, c_k^2), c_{\max} = \max(c_k^1, c_k^2), I = c_{\max} - c_{\min}$.

Мутация. Оператор мутации получает на вход одну особь, выбранную по принципу «рулетки» и меняет один из ее параметров на случайную величину по следующему правилу: $h_i = h_i + \eta(h_{\max} - h_{\min})$, где h_i – выбранный для изменения параметр, η – случайное число из интервала $[-0.0125, 0.0125]$. Границы указанного интервала могут являться параметрами алгоритма, но в данной работе они были зафиксированы.

Результатом работы оператора также является одна особь.

Вычисление приспособленности

Для вычисления приспособленности особи движение моделируется в симуляторе *Webots*. Робот проигрывает полученное движение до тех пор, пока оно не закончится, или он не упадет. После этого вычисляется функция приспособленности данного движения. Она записывается в другой файл, который считывается и обрабатывается тестером движений.

Существуют различные функции приспособленности. В данной работе использовалась функции для максимизации скорости движения (или, что эквивалентно, расстояния, проходимого за фиксированное время):

$$F(\text{individual}) = \begin{cases} 100(x_{\text{after}} - x_{\text{before}}), & x_{\text{after}} > x_{\text{before}}, \\ 50(x_{\text{after}} - x_{\text{before}}), & x_{\text{after}} > x_{\text{before}}, \\ 0, & x_{\text{after}} \leq x_{\text{before}} \end{cases}$$

При этом первая формула использовалась для устойчивого движения, а вторая – в случае падения. x_{before} и x_{after} – координаты робота (в метрах) до и после выполнения движения. Они отложены по оси, совпадающей с направлением движения вперед.

Среднее время обработки одного поколения из 30 особей составляет примерно две – четыре минуты. В процессе эволюции время, затрачиваемое на одну особь, увеличивается, так как появляется больше движений, выполняя которые робот не падает.

Отбор особей для следующего поколения

После того, как для всех особей очередного поколения вычислена их приспособленность, необходимо заполнить новое поколение.

Для этого сначала выбирается k наиболее приспособленных особей из предыдущего поколения, которые переносятся в следующее поколение в неизменном виде (элитизм).

Далее по принципу рулетки отбираются особи для кроссовера и мутации. В результате промежуточное поколение, включающее в себя предыдущее поколение и созданных потомков. Недостающие особи для нового поколения добиваются из промежуточного.

Вероятность использования мутации равна p (параметр алгоритма), $0 \leq p \leq 1$, скрещивания, соответственно, $1 - p$.

Критерий останова

Во всех экспериментах, проводимых в рамках написания данной работы, останов осуществлялся, в случае если максимальная приспособленность не менялась на протяжении 30 поколений.

Анализ полученных результатов

Был проведен ряд экспериментов для определения оптимальных параметров генетического алгоритма, таких как размер поколения, доля кроссовера и мутации в формировании нового поколения и т. п. Затем было произведено сравнение движения, полученного предложенным методом с использованием выбранных оптимальных параметров, с движением из файла `Forwards.motion`, поставляемого вместе с симулятором *Webots* в качестве примера. Результаты приведены в табл. 2.

Таблица 2. Пример задания движения в виде числовой последовательности

Движение	Скорость, м/мин	Отклонение в сторону, м/мин	Поворот в сторону, °/мин
Стандартное движение	1.64	0.15	4.43
Начальное движение	0.47	0.02	3.65
Полученное движение	2.11	0.19	0.68

Таким образом, полученное предложенным методом движение, не уступает аналогам, построенным другими методами, а по скорости — превосходит примерно на 30%.

Заключение

В работе предложен метод генерации числовых последовательностей, описывающих шаг вперед человекоподобного робота, основанный на использовании генетических алгоритмов. Описанный метод, в отличие от существующих методов генерации движений, не требует специфических знаний о физических свойствах робота, не использует специфических для данной модели робота свойств и может быть применен и для других человекоподобных роботов.

Выполнено сравнение стандартного шага вперед, описание которого распространяется вместе с симулятором *Webots*, движения построенного вручную, но с использованием описанного в работе метода, и движения построенного автоматически.

Список источников

1. <http://www.robocup.org>
2. <http://www.robotstadium.org>
3. Гладков Л. А., Курейчик В. В., Курейчик В. М. Генетические алгоритмы. М.: Физматлит, 2006.
4. Mitchell M. An Introduction to Genetic Algorithms. MIT Press. Cambridge. MA, 1996.
5. Morimoto J., Endo G., Nakanishi J. *u dp*. Modulation of simple sinusoidal patterns by a coupled oscillator model for biped walking.
<http://www.andrew.cmu.edu/user/darrinb/papers/icra2006jm.pdf>
6. Van den Kieboom J. Biped Locomotion and Stability a Practical Approach.
<http://scripties.fwn.eldoc.ub.rug.nl/FILES/scripties/Kunstmatigeintellige/Master/2009/Kieboom.J.van.den./AI-MAI-2009-J.VANDENKIEBOOM.pdf>
7. Мандриков Е. А., Кулев В. А. Разработка инструментального средства для генерации конечных автоматов с использованием генетических алгоритмов

//Научно-технический вестник СПбГУ ИТМО. Вып. 53. Автоматное программирование. 2008, с.100–103.

8. *Herrera F., Lozano M., Verdegay J. L.* Tackling real-coded Genetic algorithms: operators and tools for the behaviour analysis.

<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.56.7704>