Biped locomotion methodologies applied to humanoid robotics

Hugo Picado\textsuperscript{1,2}, Nuno Lau\textsuperscript{1,2}, Luis Paulo Reis\textsuperscript{3,4}, Marcos Gestal\textsuperscript{5} \\
hugopicado@ua.pt, nunolau@ua.pt, lpreis@fe.up.pt, mgestal@udc.es

\textsuperscript{1} Institute of Electronics and Telematics Engineering of Aveiro, Portugal \textsuperscript{2} Dep. of Electronics, Telecommunications and Informatics, Univ. Aveiro, Portugal \textsuperscript{3} Artificial Intelligence and Computer Science Lab., Univ. Porto, Portugal \textsuperscript{4} Faculty of Engineering of the University of Porto, Portugal \textsuperscript{5} Artificial Neural Network and Adaptive System Lab., Univ. Coruña, Spain

Abstract. Controlling a biped robot with several degrees of freedom is a challenging task that takes the attention of several researchers in the fields of biology, physics, electronics, computer science and mechanics. For a humanoid robot to perform in complex environments, fast, stable and adaptable behaviors are required. Developing robust behaviors requires the development of methods for joint trajectory planning and low-level control. Several methods are part of the state of the art, including trajectory-based methods, virtual model control, passive-dynamic walking and central pattern generators. This paper proposes a solution for automatic generation of a walking gait using genetic algorithms. The experimental results are shown in terms of the evolution of the ground projection of the center of mass, walking velocity and average oscillation of torso. Genetic algorithms proved to be a powerful method for automatic generation of humanoid behaviors. The robot was able to reach a walk forward velocity of 0.51m/s which is a good result considering the results of the three best teams of RoboCup 3D simulation league for the same movement.

Key words: biped, locomotion, genetic algorithms, humanoid, robotics

1 Introduction

For a long time, wheeled robots were used for research and development in the field of Artificial Intelligence and Robotics and many solutions were proposed [1]. However, wheeled robot locomotion is not adapted to many human environments [2]. This increased the interest in other types of locomotion like biped locomotion and especially in humanoid robotics. This field has been studied over the last years and many different approaches have been presented, although the ability for robots to walk in unknown terrains is still in a young stage. Several approaches to biped locomotion have been developed. This section explains the most common methods which may be broadly divided in four main categories: trajectory-based approaches, virtual model control, passive-dynamic walking and central pattern generators.
1.1 Trajectory-based approaches

Trajectory-based methods consist of finding a set of kinematics trajectories and using a stabilization criterion to ensure that the gait is stable. The most popular stabilization criteria are the Center of Mass (CoM), Center of Pressure (CoP) and Zero Moment Point (ZMP). The gait is stable when one of these criteria remains inside the support polygon (the convex hull formed by the contact points of the feet with the ground).

**Center of Mass** The CoM of a system of particles is the point at which the mass of the system acts as if it was concentrated [3]. In other words, CoM is defined as the location of the weighted average of the system individual mass particles, as defined by the following equation:

\[ p_{CoM} = \frac{\sum_i m_i p_i}{M} \]  \(1\)

where \( M = \sum_i m_i \) is the total mass of the system, \( m_i \) denotes the mass of the \( i^{th} \) particle and \( p_i \) denotes its centroid.

**Center of Pressure** Most humanoid robots are equipped with force-torque-sensors at the feet of the robot. The Center of Pressure (CoP) is the result of an evaluation of those sensors and is defined as the point on the ground where the resultant of the ground reaction forces acts [4]:

\[ p_{CoP} = \frac{\sum_i p_i F_{N,i}}{\sum_i F_{N,i}} \]  \(2\)

where the resultant force \( F_R = \sum_i F_{N,i} \) is the vector from the origin to the point of action of force \( F_{N,i} = |F_{N,i}| \).

**Zero Moment Point** The Zero Moment Point (ZMP) is perhaps the most popular stability criterion and was originally proposed by Vukobratovic [5] in 1972. It can be defined as the point on the ground about which the sum of the moments of all the active forces equals zero [5]. An alternative, but equivalent, interpretation was given by Arakawa and Fukuda [6] (See Figure 1). They define ZMP as the point \( p \), where \( T_x = 0 \) and \( T_y = 0 \), where \( T_x \) and \( T_y \) represent the moments around the x and y axis generated by the reaction force \( R \) and reaction torque \( M \), respectively.

**Static vs. dynamic stability** The static stability criterion prevents the robot from falling down by keeping the CoM inside the support polygon by adjusting the body posture very slowly thus minimizing the dynamic effects [7] and allowing the robot to pause at any moment of the gait without falling down. Using this criterion will generally lead to more power consumption since the robot has to adjust its posture so that the CoM is always inside the support polygon. On
the other hand, humans move in a dynamic fashion, a state of constant falling [7], where the CoM is not always inside the support polygon. While walking, humans fall forward and catch themselves using the swinging foot while continuing to walk forward, which makes the CoM moves forward without expending energy to adjust the CoM trajectory. Dynamic stability relies on keeping the ZMP or CoP inside the support polygon and this is a necessary and sufficient condition to achieve stability. Dynamic balance is particularly relevant during the single support phase, which means that the robot is standing in only one foot. This generally leads to more fast and reliable walking gaits.

1.2 Virtual model control

The most important drawback of ZMP is the use of complex dynamic equations to compute the robot’s dynamics. This complexity can be crucial when designing humanoid robots, specially when the programmer wants to minimize the power and memory consumption of the biped. Developed by Jerry Pratt [8], Virtual Model Control (VMC) is a framework based on heuristics that uses virtual components such as springs, dampers or masses to generate the joint torques that control the biped’s stability and velocity. The generated joint torques create the same effect that the virtual components would create if they were in fact connected to the real robot. This heuristic makes the design of the controller much easier. First it is necessary to place some virtual components to maintain an upright posture and ensure stability. Using the example provided by Pratt [8], imagine that the goal of the robot is to knock a door. With VMC it is only needed to place a virtual mass with a specified kinetics energy to the robot’s hand using a virtual spring and damper. The robot’s hand will then move to strike out and once given the desired impact, the hand will get back due to mass resonating with the virtual component attached to the hand.

1.3 Passive-dynamic walking

In the Passive-Dynamic Walking (PDW) approach, the biped walks down a slope without using any actuator [9]. The mechanical characteristics of the legs (e.g. length, mass, foot shape) determine the stability of the generated walking motion.
PDW is based on the inverted pendulum model [9], which assumes that, in the single support phase, human walking can be modeled as an inverted pendulum. Inverted pendulum has been applied for years in several situations [11]. The swinging leg (assuming there is just the hip and the ankles and no knees) is represented by a regular pendulum, while the support leg is represented by an inverted pendulum. The support leg is then controlled by the hip joint’s torque. However, in the specific case of PDW the only actuating force is the gravity. Figure 2 represents the PDW model. Tad McGeer, in 1990, was the first to apply this idea to humanoid robotics by developing a 2D bipedal robot with knee joints and curved feet [9]. The developed robot was able to walk down a three degree slope. This work demonstrated that the morphology of the robot might be more important than the control system itself. This method is known for the low power consumption.

1.4 Central pattern generators

It is assumed, by the fields of biology, that vertebrate locomotion is controlled by a spinal central pattern generator (CPG) [12]. A CPG is a set of circuits which aims to produce rhythmic trajectories without the need for any rhythmic input. In legged animals, the CPG often contains several centers that control different limbs. There has been a growing interest in these CPG models in robotics. This trajectory planning method does not need, necessarily, any sensory feedback information to generate oscillatory output for the motor neurons [12]. However, it is possible to integrate the sensory feedback information such as force resistors and gyroscopes to produce motion correction and compensation [13].

By coupling the neural oscillators signals when they are stimulated by some input, they are able to synchronize their frequencies. In the field of artificial intelligence and robotics, it is possible to build structures that are similar to the neural oscillators found in animals by the definition of a mathematical model. However, most of the times these CPGs are designed for a specific application and there are very few methodologies to modulate the shape of the rhythmic signals, in particular for online trajectory generation [14], which lead the researchers to use other methods as the ones presented previously. Sven Behnke [15] proved...
that it is possible to apply CPGs to generate an omnidirectional walking gait, where the input is simply the forward, lateral and rotational walking speed.

The solution presented in this paper for joint trajectory generation has the smooth properties of the central pattern generators since it is based on oscillators but on the other hand can be more easily defined.

2 Genetic algorithms

Proposed by the mathematician John Holland in 1975 [16], A Genetic Algorithm (GA) is an optimization method inspired by the evolution of biological systems and based on global search heuristics. GA belongs to the class of evolutionary algorithms. In spite of being different, evolutionary algorithms share common properties since they are all based on the biological process of evolution. Given an initial population of individuals (also called chromosomes), the environmental pressure causes the best fitted individuals to survive and reproduce more. Each individual (chromosome) is a set of parameters (genes) and represents a possible solution to the optimization problem. The algorithm starts by creating a new population of individuals. Typically, this population is created randomly but any other creation function should be acceptable. The genes of each individual should be inside a range of acceptable values that is defined for each gene. The algorithm then starts the evolution which consists of creating a sequence of new populations. At each step, the algorithm uses the individuals in the current population to create the next population by applying several operators. These genetic operators are described as follows [17]:

- **Selection**: Chooses some parents for crossover according to predefined rules (cost function or fitness);
- **Elitism**: Defines the number of chromosomes in the current generation that are guaranteed to survive in the next generation;
- **Crossover**: Generates offspring by exchanging some genes of different parent chromosomes;
- **Mutation**: Generates a mutant gene by changing one or several genes of an individual.

In the end of the optimization process, the individual in the current population that have the best fitness value is chosen as the best individual. Algorithm 1 shows the pseudo code of a generic GA.

There are many alternatives to the use of GAs for gait optimization. Some of these alternatives might be Hill Climbing (HC) [18], Simulated Annealing (SA) [19], Tabu Search ([20], or even machine learning methods such as Reinforcement Learning (RL) [21]. The option for GA in the present work is due to the good results obtained in previous experiments in the same research field [22].
Algorithm 1 Generic Genetic Algorithm

\begin{algorithm}
\hspace{1em}Population ← \text{CreateInitialPopulation()}
\hspace{1em}Evaluate(Population)
\hspace{1em}while \text{TerminationConditionNotMet()} do
\hspace{2em}[Selection] Parents ← Selection(Population)
\hspace{2em}[Elitism] Elite ← Elitism(Population)
\hspace{2em}[Crossover] Children ← Crossover(Parents, p_c)
\hspace{2em}[Mutation] Mutants ← Mutation(Children, p_m)
\hspace{2em}Population ← Elite + Mutants
\hspace{2em}Evaluate(Population)
\hspace{1em}end while
\hspace{1em}return Best(Population)
\end{algorithm}

3 Walking gait definition

A walking gait was developed and the tests were performed with the simulated humanoid NAO in the scope of the RoboCup 3D Soccer competition using the Simspark Simulation Environment [23]. Figure 3 shows the humanoid structure and the referential axis considered.

Fig. 3: Humanoid structure and global referential. The arrows around the axes represent the positive direction of the pitch, roll and yaw rotations. Adapted from [24].
3.1 Joint trajectory planning

The method used for joint trajectory planning is based on Partial Fourier Series (PFS). Some human-like movements are inherently periodic and repeat the same set of steps several times (e.g. walk, turn, etc). The principle of PFS consists of the decomposition of a periodic function into a sum of simple oscillators as represented by the following expression:

\[ f(t) = C + \sum_{n=1}^{N} A_n \sin \left( \frac{2\pi}{T} t + \phi_n \right), \forall t \in \mathbb{R}_0^+ \]  

(3)

where \( N \) is the number of frequencies, \( C \) is the offset, \( A_{n=1..N} \) are amplitudes, \( T \) is the period and \( \phi_{n=1..N} \) are phases.

The main idea behind the definition of the walking gait is to place an oscillator on each joint we pretend to move in order to define its trajectory. The oscillators are placed on the following joints: LShoulder1, RShoulder1, LThigh1, RThigh1, LThigh2, RThigh2, LKnee, RKnee, LAnkle1, RAnkle1, LAnkle2 and RAnkle2. Hence, 12 single-frequency oscillators are used. Since each single-frequency oscillator will have 4 parameters to define, 48 parameters are needed to completely define the gait. It is common to assume a walk sagittal symmetry, which determines the same movements for corresponding left and right sided joints with a half-period phase shift. Hence, it is possible to reduce the number of parameters by half of the original size, resulting on 24 parameters. Additionally, the period of all oscillators should be the same to keep all the joints synchronized by a single frequency clock. This consideration reduces the number of parameters to 19. A set of equations can be obtained for the left-sided joints:

\[ f_{LShoulder1}(t) = C_1 + A_1 \sin \left( 2\pi t/T + \phi_1 \right) \]  

(4)

\[ f_{LThigh1}(t) = C_2 + A_2 \sin \left( 2\pi t/T + \phi_2 \right) \]  

(5)

\[ f_{LThigh2}(t) = C_3 + A_3 \sin \left( 2\pi t/T + \phi_3 \right) \]  

(6)

\[ f_{LKnee}(t) = C_4 + A_4 \sin \left( 2\pi t/T + \phi_4 \right) \]  

(7)

\[ f_{LAnkle1}(t) = C_5 + A_5 \sin \left( 2\pi t/T + \phi_5 \right) \]  

(8)

\[ f_{LAnkle2}(t) = C_6 + A_6 \sin \left( 2\pi t/T + \phi_6 \right) \]  

(9)

where \( f_X(t) \) is the trajectory equation for the joint \( X \), \( A_{i=1..6} \) are amplitudes, \( T \) is the period, \( \phi_{i=1..6} \) are phases and \( C_{i=1..6} \) are offsets. The right-sided joints can be obtained with no additional parameters: For roll joints the left and the right side perform the same trajectories over the time. For pitch joints, the right side can be obtained by adding a phase, \( \pi \), on the corresponding oscillator. The unknown parameters together form the genome that will be used by the genetic algorithm to generate the gait.
3.2 Automatic generation of parameters

The parameters described in the previous section were defined by a GA. The algorithm was configured using an initial population of 100 chromosomes initialized randomly. The roulette method used for selection consists of simulating a roulette-wheel where the parents are selected with a probability that is proportional to their fitness. The mutation follows an uniform distribution with a probability defined by $p_m = 0.5$. For crossover the scattered method was used. Scattered creates a random binary vector and selects the genes where the vector is a 1 from the first parent, and the genes where the vector is a 0 from the second parent. The fraction of the population that is created by crossover is defined by the parameter $p_c = 0.8$. For the elitism, 10 chromosomes are selected to survive for the next generation.

The fitness function has to be chosen carefully in order to achieve good results. In the case of the forward walking, a simple but effective fitness function to minimize can be the distance to some point in the forward direction (let’s call it target), assuming that the robot is initially placed far enough from it. Additionally, the torso average oscillation [25] is also used in order to obtain more stable gaits. The final version of the fitness function is stated as follows:

$$\text{fitness} = d_{\text{target}} + \bar{\theta}$$ (10)

where $d_{\text{target}}$ is the distance to the target point (in meters) and $\bar{\theta}$ is the average oscillation of the torso (in radians per second). The generation process took five entire days to complete using a Core 2 Duo 2.4Gz CPU with 1GB of physical memory. Figure 4 shows the evolution of the fitness during the optimization process.

The fitness decreases fast and stabilizes in a hundred of generations. The minimum fitness is 0.20311, which is a very good result. Table 1 shows the values of the best individual for $A_{1,6}$, $\phi_{1,6}$ and $C_{1,6}$. For the period, $T$, the optimal generated value was 0.3711.
Table 1: Best generated individual

<table>
<thead>
<tr>
<th></th>
<th>A_1</th>
<th>φ_1</th>
<th>C_1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>57.1842</td>
<td>2.9594</td>
<td>-88.4624</td>
</tr>
<tr>
<td>2</td>
<td>5.6445</td>
<td>-2.2855</td>
<td>3.6390</td>
</tr>
<tr>
<td>3</td>
<td>57.1211</td>
<td>0.0887</td>
<td>-39.9481</td>
</tr>
<tr>
<td>4</td>
<td>39.6205</td>
<td>-1.8292</td>
<td>28.5095</td>
</tr>
<tr>
<td>5</td>
<td>46.6315</td>
<td>1.7640</td>
<td>-3.9481</td>
</tr>
<tr>
<td>6</td>
<td>3.7947</td>
<td>-1.2067</td>
<td>-2.9360</td>
</tr>
</tbody>
</table>

Figure 5 shows the evolution of the CoM and the placement of feet in the XY plane. It is possible to note that the robot tends to shift the CoM to the support foot while walking. Another characteristic shown by the same graphic is the large size of the steps.

The average velocity (Figure 6a) shows very good results. More than 50 centimeters per second were achieved. This is a good velocity taking into account the torso average oscillation, that is represented in the Figure 6b.
The obtained results can be considered good results, comparing with the three best teams of the RoboCup 3D simulation league competition of 2008 (Suzhou, China). They were able to reach forward velocities of 1.20m/s (SEU-RedSun [26]), 0.67m/s (WrightEagle [27]) and 0.43m/s (LittleGreenBats [28])\(^6\). Figure 7 shows NAO walking forward using the proposed solution. At t=1.44s the biped already covered a great distance. It is also possible to note the large steps and the height of the steps.

![Walking gait screenshots](image)

Fig. 7: Walking gait screenshots

4 Conclusion and future work

This paper presented the common approaches to biped locomotion. These approaches were broadly divided in trajectory-based approaches which consist of keeping some stabilization criteria inside the support polygon of the humanoid, the virtual model control which is based on heuristics aiming at reducing the computational effort, the passive-dynamic walking where a robot walks down a slope using only the gravitational force and, finally, the central pattern generators that pretend to imitate the locomotion mechanism that is assumed to exist in vertebrates. It was presented a solution for automatic generation of a forward walking gait using a genetic algorithm. GAs are based on the biological evolution of species by applying selection, mutation and crossover operators to a set of chromosomes that are formed by genes representing the parameters to be optimized. GA proved to be very good on achieving results. The generated walking gait is fast, stable and resistant to environment disturbances. These results were successfully compared to the results of the three top teams of 3D simulation league of RoboCup in 2008.

---

\(^6\) This results were retrieved from the logfiles of the RoboCup 2008 competition, which may be found in [http://www.robocup-cn.org/](http://www.robocup-cn.org/).
Several improvements are possible and needed. The generated walk was mainly based on the trajectory of CoM to monitor the quality of the gait. However, the calculation and monitoring of the ZMP trajectory is essential for achieving dynamic stability. Additionally, the use of inverse kinematics to compute the trajectories of end effectors instead of controlling the joints directly is very useful to have more flexibility among the generation of behaviors. Moreover, motion capturing, which consists of monitoring the human behaviors, provides a great way to define the humanoid behaviors, due to the anthropomorphic characteristics between the both. As future work, it is predicted to invest not only in genetic algorithms, but in machine learning methods such as reinforcement learning. These methods provide great advantages for the automatic generation of behaviors which reduce, and possibly eliminate, the human intervention during the optimization or learning process.

Acknowledgements

This research was partially supported by FCT-PTDC/EIA/70695/2006 Project – "ACORD - Adaptive Coordination of Robotic Teams".

References


