The Walking Skill of Apollo3D – The Champion Team in the RoboCup2013 3D Soccer Simulation Competition

Juan Liu, Zhiwei Liang, Ping Shen, Yue Hao, and Hecheng Zhao

College of Automation, Nanjing University of Posts and Telecommunications, Nanjing, 210046

Abstract. Quick and flexible walking is an indispensable skill for humanoid robots in the RoboCup soccer competition. So this paper mainly proposed a method to develop a flexible walking based on reinforcement learning for humanoid robots, which used Cerebellar Model Articulation Controller(CMAC) method and a linear inverted pendulum with a predictive control to generate a motion trajectory of the robots trunk in the premise of keeping dynamic balance of robots. Our team Apollo3D employed this walking skill, and won the championship in the RoboCup 2013 3D simulation competition.

Keywords: Cerebellar model articulation controller, preview control, linear inverted pendulum, trunk trajectory.

1 Introduction

How to make humanoid robots play soccer like human is a big challenge in the fields of artificial intelligence and intelligent robotics. It provides a benchmark platform for the state of the art in science and technology. Nowadays, the international competitions of robot soccer have become more and more common, the most influential of which is the Robot World Cup (RoboCup) and its ultimate goal is to develop a team of fully autonomous humanoid robots that can win against the human world soccer champion team in 2050. In order to complete the goal, different types of robot soccer league have been organized in the international and domestic, and the simulation game is the important one. The RoboCup 3D Simulated Soccer League allows external processes to control humanoid robots to compete against one another in a realistic simulation of the rules and physics of a soccer game. The simulation is executed in the RoboCup Simulated Soccer Server 3D (rcssserver3d) which runs on Linux, Windows and MacOS X. The underlying simulation engine is SimSpark.Since 2012, RoboCup3D simulation team is composed of eleven autonomous robots (NAO robots from Aldebaran) and each robot can communicate with other robots through the network.

In order to perform a fluent soccer match for humanoid robot teams, various basic actions must be implemented, including walking, kicking, rising from

S. Behnke et al. (Eds.): RoboCup 2013, LNAI 8371, pp. 104–113, 2014.

falling, and so on. Among these robotic skills, a steady dynamic walking is the most foundamental factor. Many studies have been proceeding in the walking of humanoid robot, with different degree of success. For example, Storm et al[1]. presented the concept of joint trajectory planning, which analyzed the smooth hip motion with the largest stability margin and derived the highly stable hip trajectory by iterative computation without calculating the desired ZMP (zero moment point) trajectory. This method is also suitable for different ground conditions by varying the values of the foot constraint parameters. However, the trajectory tracking is generated offline and computationally intensive. In order to make online feedback possible, Bavani et al. put forward a method called central pattern generator (CPG) [2-3], which is based on the neural network method. CPG is a circuit system, which can create a self-contained periodic signal and be initialized by the non-oscillation signal. The most difficulty in realizing this method is how to determine the appropriate weights of neural connections. Another approach by incorporating feedback into walking is the passive dynamic walking. Seung-Joon showed that a robot could walk down a slope without any actuator and control[4]. The operating forces need in walking lessened by gravity and swing in its natural frequency. This method can be applied to walk on a flat ground by relying on foot contact sensors.

Since the above methods only focus on stable biped movements in a particular period without high efficiency, they are not suitable for the soccer match environments with dynamic confrontions and limited space. To improve the walking skill of biped robots, it is important to realize a reinforcement learning walking method [5]. So this paper describes a robust reinforcement learning walking method which based on Cerebellar Model Articulation Controller(CMAC)[6-7].

2 CMAC Walking Method

The primary key to a successful robot team in the RoboCup3D simulation soccer competition was to realize the robots steady and robust gait, as well as to realize fully stable walking. The main advantage of this walking is that a robot can maintain moving forward, side and turning around in the premise of keeping stable velocities on approaching the destination.

In the process of competition, the changing external environment requires the robot to alter its orientation at any time, turn agilely and forward fast. The CMAC walking method is presented in the Fig.1. First we can get the feasible footholds and compute the ZMP values based on the foot-planning module. Subsequently, the trunk trajectory of robot can be attained based on a double linear inverted pendulum model (D-LIP) with a predictive control method[8]. As a result, we can plan the space trajectory of every two footholds in 3D space according to the cubic spline interpolation method[9]. Meanwhile, each joint angle can be calculated according to inverse kinematics method[10]. The pose of the robots trunk can easily computed by the gyro sensor of NAO[11]. Last but not least, we use CMAC optimization and correction algorithm to control the movement of the leg joint accurately. And the whole system is forming a feedback control with the D-LIP.

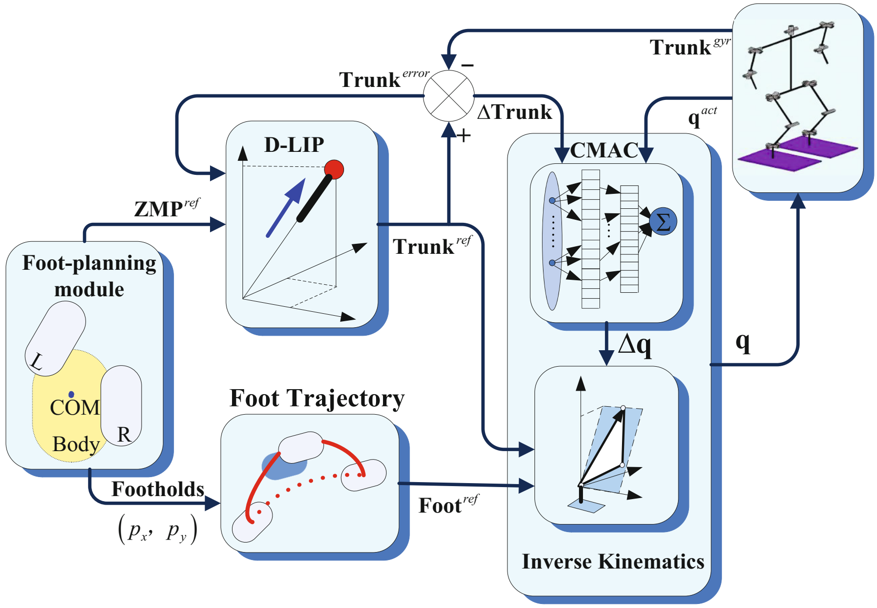


Fig. 1. Control block diagram of the CMAC walking

2.1 Foot-Planning Module

In the process of competition, players uses the designation of the footholds $(P_0, P_1, P_2, P_3 \cdots P_n)$ to plan each step of the robot. As shown in Fig.2, define $S_x^{(n)}$ as the step size in forward direction,$S_y^{(n)}$ as the step width in lateral direction,θ as the orientation of each step, superscript (n) as the nth step, and$(S_x^{(n)}, S_y^{(n)}, \theta)$ are called walking parameters. The nth foothold can be expressed as:

$$\begin{pmatrix} p_x^{(n)} \\ p_y^{(n)} \end{pmatrix} = \begin{pmatrix} p_x^{(n-1)} \\ p_y^{(n-1)} \end{pmatrix} + \begin{pmatrix} c_\theta^{(n)} & -s_\theta^{(n)} \\ s_\theta^{(n)} & c_\theta^{(n)} \end{pmatrix} \begin{pmatrix} s_x(n) \\ -(-1)^n s_y(n) \end{pmatrix} \tag{1}$$

where $c_\theta = cos_\theta, s_\theta = sin_\theta, (p_x^{(0)}\ p_y^{(0)})$ is the position of the first support leg and assume the left foot is the beginning foot (if not, then the $-(-1)^n$ will be turned into $(-1)^n$).

For the walking trajectory, every step of the 3D-LIPM track called walking unit is a hyperbolic in the x-y plane. Then the nth step walk unit $(\bar{x}^{(n)}, \bar{y}^{(n)})$ is determined by the following walking parameters:

$$\begin{pmatrix} \bar{x}^{(n)} \\ \bar{y}^{(n)} \end{pmatrix} = \begin{pmatrix} c_\theta^{(n+1)} & -s_\theta^{(n+1)} \\ s_\theta^{(n+1)} & c_\theta^{(n+1)} \end{pmatrix} \begin{pmatrix} s_x^{(n+1)}/2 \\ (-1)^n s_y^{(n+1)}/2 \end{pmatrix} \tag{2}$$

Given the supporting time of each step T_s, the altitude position of the center of mass (COM)z_c , the norm of the gravitational force g, the termination speed of the walking unit can be expressed as:

$$\begin{pmatrix} \bar{v}^{(n)} \\ \bar{v}^{(n)} \end{pmatrix} = \begin{pmatrix} c_\theta^{n+1} & -s_\theta^{n+1} \\ s_\theta^{n+1} & c_\theta^{n+1} \end{pmatrix} \tag{3}$$

where $T_c = \sqrt{Z_c/g}, C = cosh(T_s/T_c), S = sinh(T_s/T_c)$.

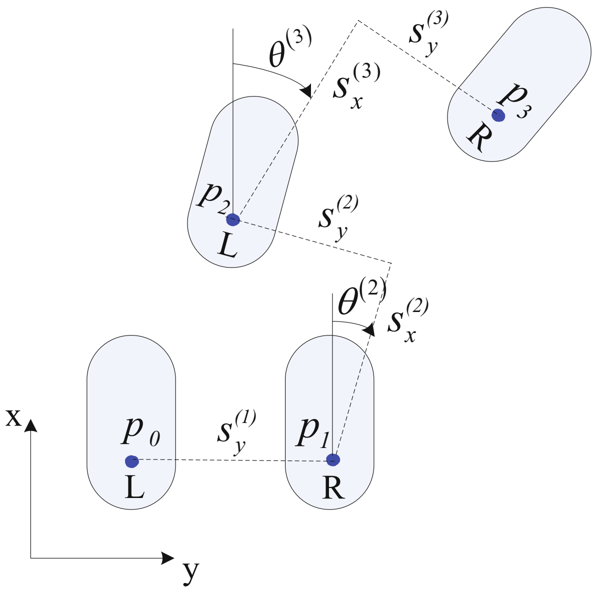


Fig. 2. The foothold planning

2.2 Based on the ZMP Predictive Control of D-LIP

The trajectory of the COM and the ZMP are decentralized by the sampling time t through a cubic polynomial. Apollo3D use a preview control method based on the trajectory of COM to predict the ZMP, and the N sample point values of the ZMP are need to calculate the current value of COM at the same time.

$$\mathbf{X}_{ZMP}(k+1) = \mathbf{A} \cdot \mathbf{X}_{COM}(k) + \mathbf{B} \cdot \dddot{\mathbf{X}}_{COM}(k) \tag{4}$$

with $\mathbf{X}_{ZMP}(k+1) = \begin{pmatrix} x_{ZMP}(k+1) \\ \vdots \\ x_{ZMP}(k+N) \end{pmatrix}, \dddot{\mathbf{X}}_{COM}(k) = \begin{pmatrix} \dddot{x}_{COM}(k) \\ \vdots \\ \dddot{x}_{COM}(k+N-1) \end{pmatrix}$

$$\mathbf{A} = \begin{pmatrix} 1 & t & t^2/2 - z_c/g \\ \vdots & \vdots & \vdots \\ 1 & Nt & N^2t^2/2 - z_c/g \end{pmatrix}, \mathbf{B} = \begin{pmatrix} b_0t^3/6 - tz_c/g & 0 & 0 \\ \vdots & \ddots & 0 \\ b_{N-1}t^3/6 - tz_c/g & \cdots & b_0t^3/6 - tz_c/g \end{pmatrix}$$

where z_{ZMP} denotes the horizontal displacement of the ZMP , $b_i = (1+3i+3i^2)$ in the matrix B.

2.3 Foot Trajectory

The foot positions of every walk cycle are directly deduced from the foot-planning module. These footholds (p_x, p_y) are used to compute the continuous curve of foot movement. Due to a large amount of calculation, a violent oscillation and the poorly numerical stability about the high order function, Apollo3D employed the cubic spline interpolation method to compute the foot trajectory to solve all of the above problems.

2.4 CMAC Optimization and Correction Algorithm

CMAC is used as the walking leg controller of a robot to stabilize inverse kinematics model from position to joint space mapping. The trunk motion trajectory is defined as the given signal $Trunk_{t+1}^{ref}$ at time t+1, comparing with the real trunk position $Trunk_t^{gyr}$ at time t, we can obtain the position increment $\Delta\mathbf{Trunk}_{t+1}$, which combined with the leg angle $\mathbf{q}^{act}$ as the input of this control system. And the output of CMAC neural network is $\Delta\mathbf{q}_{t+1}$ in order to control the movement of the leg joint.

Due to the solution of leg inverse kinematics is not unique, we need a joint angle optimization to select a proper output angle. This optimization problem can be solved by gradient descent iteration, the optimization goal function can be written in vector form:

$$\mathbf{Y} = \frac{\alpha}{2}\mathbf{e}^T\mathbf{e} + \frac{\beta}{2}\Delta q^T \Delta q$$

Where $\mathbf{e}$ is the error vector, Δq is the joint angle increment.

The iterative algorithm about Δq can be expressed as follows:

$$\Delta q^{k+1} = \Delta q^k - \eta\frac{\partial \mathbf{Y}}{\partial \Delta q}, (\partial \mathbf{Y}/\partial \Delta q) = -\alpha\mathbf{e}^T\mathbf{J} + \beta\Delta q^T$$

where k indicate the number of iterations, η is learning step length, $\mathbf{J}$ is the leg Jacobin matrix.

Use the leg position deviation $\mathbf{e}_{t+1}$ in t+1 time to revise the weight of the controller ω_{ij}. Define the index function V and the correction algorithm about ω_{ij}:

$$\mathbf{V} = \mathbf{e}^T\mathbf{e}/2, \omega_{ij} = \omega_{ij} - \eta\frac{\partial \mathbf{V}}{\partial \omega_{ij}}, \partial \mathbf{V}/\partial \omega_{ij} = \mathbf{e}^T(-\mathbf{J}_i)$$

where i=1,2; j=1,2,$\cdots$ C indicate the logical address; η is the positive learning ratio.

2.5 Inverse Kinematics

The solution of the inverse kinematics equation is to compute all joint angle values when given a certain position of the object. The $\mathbf{Trunk}^{ref}$ and $\mathbf{Foot}^{ref}$ are respectively deduced from the double inverted pendulum model and the foot trajectory pattern, and correction of joint angle $\Delta\mathbf{q}$ that is deduced from CMAC module would be calculated in inverse kinematics.

2.6 Feedback Control Strategies

In the dynamic model of walking, the closed loop control is mainly to consider the error of the robot trunk position $\mathbf{Trunk}^{error}$. It is not easy to get the accurate value because of the unavoidable mechanical recoil force in each legs joints[10]. As is well known[8], the knees are always bending to avoid singular attitude in the process of walking. As long as the joints do not oscillate in the recoil force zone, it has assertion that the trunk error is aroused by this force. So the trunk error is considered in the double linear pendulum ($\mathbf{Trunk}^{error} = \mathbf{Trunk}^{ref} - \mathbf{Trunk}^{gyr}$) for the correction of D-LIPM original output value. We also used the double balance control mechanism to consolidate the stability of the robot, which is the center of mass balancing and gyroscope feedback-based balancing.

3 Experimental Results

This paper is based on a simulation model of the NAO robot, which has 22 degrees of freedom(DOF) to control the motion of its body and keep the center of mass of the robot above the area support by the feet. Our algorithm is applied in computer with configuration of Intel(R) Core(TM) i7-2600 CPU @ 3.40GHZ and simulation in the following charts.

Fig.4 shows the body rotation angle. The maximum value of the body rotation angle is 1.28 degrees, and this occurred when the support leg changed. The blue trajectories in Fig.3 and Fig.4 show the hip and the body rotation movement without using any reinforcement learning, and after the 19th learning that describe in red trajectories in Fig.4. From these two figures, the robot can walk and turn more stably.

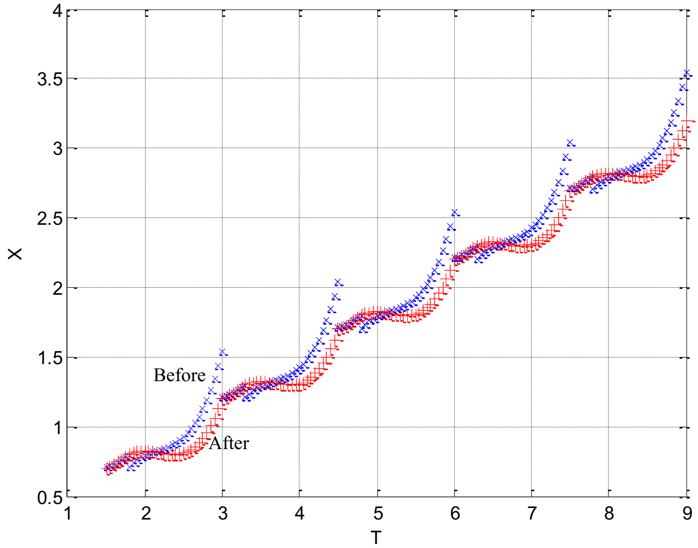


Fig. 3. Hip trajectory along the X axis

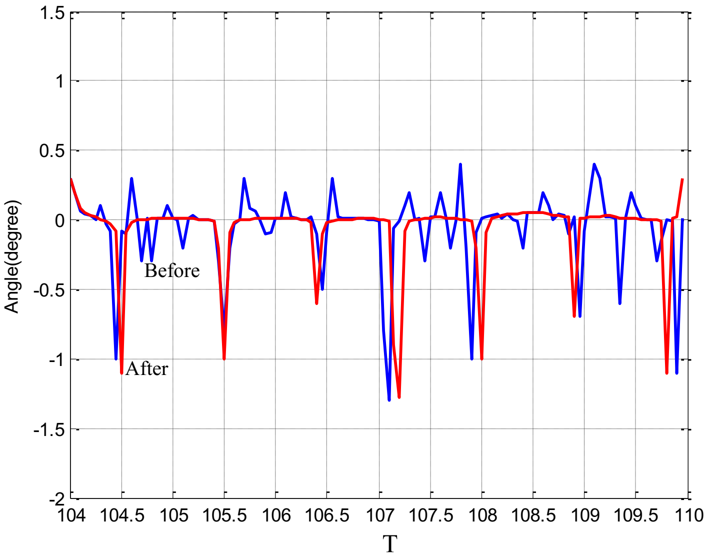


Fig. 4. Body rotation angle

Flexible turning walk for 22 steps, and simulate in the MATLAB environment is shown in Fig.5.

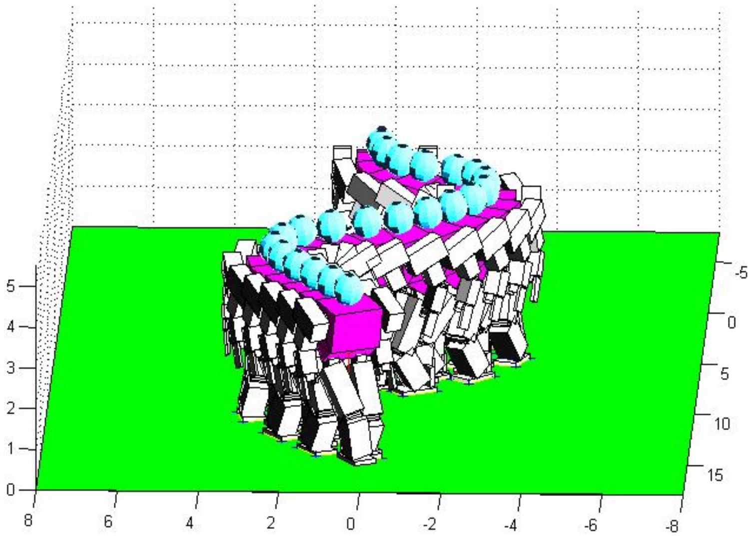
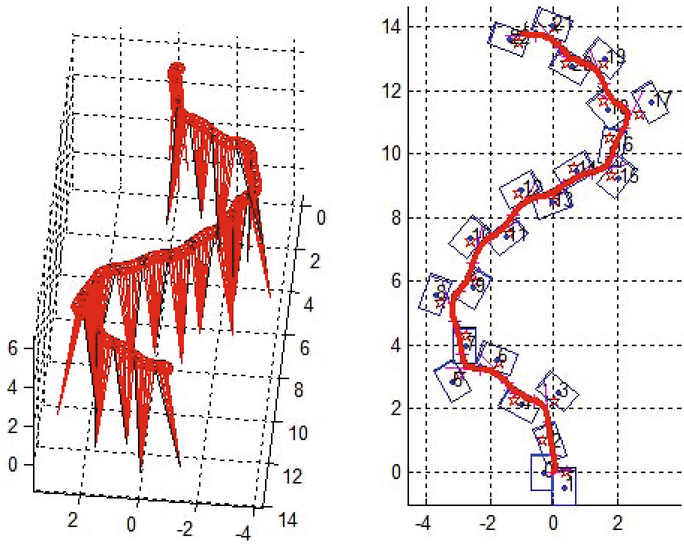


Fig. 5. Schematic diagram of CMAC walking



(a) D-LIP walking model (b) The foot trajectory

Fig. 6. The trunk trajectory diagram

Fig.6 is a top view of Fig.5, the red lines in Fig.6a show the leg walking module based on D-LIP, and the black pentagons indicate the shifting process of the center of gravity between the two legs of robot. The Fig.6b shows the robots trunk trajectory (the red curve) and the foothold positions (the blue rectangles). The blue solid point indicates the actual foothold of each step and the pentagon means the foot position of the next time by using the preview control method. From these figures, it is known that the predicted values are basically coincidence with the actual foothold positions.

In addition, we perform some experiments to verify the validity of our algorithm. It can be seen from Table 1 is the average goals (twenty games) about our algorithm (Apollo3D) vs. Apollo3D__ Before (the only difference is without using any reinforcement learning walking) and the top eight teams in RoboCup 2013 3D soccer simulation competition.

Table 1. The data contrast results based on the omnidirectional walking pattern

Rank	Team Name	Goal Diff(error)
1-2	UT__Austinvilla	-0.95 (±0.06)
3	Apollo3D__Before	1.25 (±0.13)
4-5	Fcportugal	1.32 (±0.16)
4-5	SEU__Jolly	1.53 (±0.15)
6-7	Robocanes	1.74 (±0.20)
7-8	Boldhearts	2.51 (±0.14)
7-8	Magmaoffenburg	2.06 (±0.08)
9	Hfutengine	5.21 (±0.05)

4 Conclusion

Apollo3D won 53 goals during the RoboCup 2013 3D simulation competition and lose 5 goals. Although, the results of this competition not simply rely on walking skill, but agile and stable walking ability are the foundation and the most important factor in robotic soccer.

The establishment of a powerful humanoid robot soccer team not only need some good action skills, but also the upper decision module, and also involves a lot of optimization problems. Apollo3D takes various cooperation actions with a not optimal efficiency, so the next work is that how to optimize the parameters in the actions and the upper decision modules.

Acknowledgments. This paper was supported by the Open Foundation of Jiangsu Province Science Foundation (No.BK2012832) and the Natural Science Foundation of China (No.61104216 and No. 60805032).

References

[1] Strom, J., Slavov, G., Chown, E.: Omnidirectional walking using ZMP and preview control for the nao humanoid robot. In: Baltes, J., Lagoudakis, M.G., Naruse, T., Ghidary, S.S. (eds.) RoboCup 2009. LNCS (LNAI), vol. 5949, pp. 378–389. Springer, Heidelberg (2010)

[2] Bavani, M., Ahmadi, H., Nasrinpour, R.: A closed-loop Central Pattern Generator approach to control NAO humanoid robots' walking. In: Control, Instrumentation and Automation (ICCIA), pp. 1036–1041 (2011)

[3] Behnke, S.: Online trajectory generation for omnidirectional biped walking. In: Proc. of IEEE Int. Conf. on Robotics and Automation, USA, pp. 1597–1603 (2006)

[4] Seung-Joon, Y., Byoung-Tak, Z., Hong, D., Lee, D.: Online learning of a full body push recovery controller for omnidirectional walking. In: IEEE-RAS International Conference on Humanoid Robots (Humanoids), Bled, pp. 1–6 (2011)

[5] MacAlpine, P., Urieli, D., Barrett, S., Kalyanakrishnan, S., Barrera, F., Stone, P.U.: Austin Villa 2012 3D Simulation Team report (EB/OL). Technical Report AI11-10, The Univ. of Texas at Austin, Dept. of Computer Science, AI Laboratory (December 2012)

[6] Chih-Min, L., Chih-Hsuan, C., Chih-Min, L., et al.: CMAC-based dynamic-balancing design for humanoid robot. In: Proceedings of SICE Annual Conference 2010, August 18-21, pp. 1849–1854 (2010)

[7] Lee, J., Oh, J.-H.: Biped walking pattern generation using reinforcement learning. In: 2007 7th IEEE-RAS International Conference on Humanoid Robots, November 29-December 1, vol. 1, pp. 416–421 (2007)

[8] Gouaillier, D., Kilner, C.: Omni-directional closed-loop walk for NAO. In: IEEE-RAS International Conference on Humanoid Robots (Humanoids), Nashville, TN, pp. 448–454 (December 2010)

[9] Qiang, H., Kajita, S., Koyachi, N.: A High Stability, Smooth Walking Pattern for a Biped Robot. In: Proc. of IEEE Int. Conf. on Robotics and Automation, USA, pp. 65–71 (1999)

[10] Graf, C., Hartl, A., Rofer, T., Laue, T.: A robust closed-loop gait for the standard platform league humanoid. In: Proc. of the 4th Workshop on Humanoid Soccer Robots in Conjunction with IEEE-RAS Int. Conf. on Humanoid Robots, pp. 30–37 (2009)

[11] B-Human team. B-human:Team report and code release 2011, Chapter 5 (EB/OL) (Feburary 2012), `http://www.b-human.de/downloads/bhuman11coderelease.pdf` (accessed)