Gait Optimization for Multiple Humanoid Robots Based on Parallel Multi-swarm Particle Swarm Algorithm

Chunguang Li
School of Computer and Information Engineering, Changzhou Institute of Technology
Changzhou, Jiangsu, China
leechunguang76@163.com

Rongyi He
Institute of Intelligence Science and Technology, School of Computer and Information, Hohai university
Nanjing, Jiangsu, China
hery6658@163.com

Lina Yao
School of Computer Science and Engineering, University of New South Wales
Sydney, NSW, Australia
lina.yao@unsw.edu.au

Chongben Tao
School of Electronic and Information Engineering, Suzhou University of Science and Technology
Suzhou, Jiangsu, China
tom1tao@163.com

ABSTRACT

In the RoboCup 3D simulation competition, how to find a flexible and stable gait pattern is one of the keys to win the match. To achieve such walking gait, a machine learning method of optimizing the vertical Center of Mass(CoM) trajectory is presented. The vertical CoM trajectory is planned by multiple polynomial function. Inverted Pendulum Model(IPM) and a numerical method are utilized to control the Zero Moment Point(ZMP). Then the key parameters are extracted from the gait pattern, a distributed multi-robot training environment based on RoboCup 3D simulated platform is constructed, and the parallel multi-swarm particle swarm algorithm is applied to optimize the parameters. The results of experiment and competition demonstrate that the effectiveness of the proposed method.

CCS CONCEPTS

• **Computer systems organization → Embedded systems**; *Redundancy*; Robotics;

KEYWORDS

humaniod robot; multi-swarm Particle Swarm Optimization; gait optimization; distributed system; RoboCup

ACM Reference format:
Chunguang Li, Rongyi He, Lina Yao, and Chongben Tao. 2017. Gait Optimization for Multiple Humanoid Robots Based on Parallel Multi-swarm Particle Swarm Algorithm. In *Proceedings of Mobiquitous Conference, Melbourne, VIC, AU, Nov. 2017 (MOBIQUITOUS2017),* 9 pages.
https://doi.org/10.1145/3144457.3144466

MOBIQUITOUS2017, Nov. 2017 , Melbourne, VIC, AU

ACM ISBN 978-1-4503-5368-7/17/11…$15.00
https://doi.org/10.1145/3144457.3144466

1 INTRODUCTION

In recent years, many researchers have proposed a variety of effective and stable biped walking algorithms [1]. However, it is still one of the key issues in the field of biped robot research about how to achieve a stable and fast biped walking. The Robot World Cup(RoboCup) is an annual international robotics competition. It aims to promote Robotics and AI research, education and applications through competition [2]. Biped walking with a flexible and stable gait pattern in a complex dynamic environment plays a very important role in humaniod robot soccer competition.

Biped walking of humanoid robot has been extensively studied by means of simple models. The 3-D linear inverted pendulum model (3-D LIPM) is one of the most widely used simplified dynamic models of humanoid robots [3, 4]. Graf et al. [5] presented a closed-loop 3D-LIPM gait for the RoboCup Standard Platform League Humaniod and realized omni-directional walking on NAO biped robot. Guo et al. [6] extended 3D-LIPM using virtual ground method and achieved stable and dynamical omni-directional walking on a slope. In [7], an optimal preview controller based on 3D-LIPM was proposed. In [8], Zhang and Zhou introduced innovative geodesic equations based on 3D-LIPM in spherical coordinate system. After set the CoM height as a constant value, linear motion equations of CoM can be deduced based on 3-D LIPM, but the Walking based on 3-D LIPM is speed limited and not humanlike. Biomechanical research has shown that the height of CoM is variable during human walking [9], therefore some researchers have studied biped walking with variable CoM height. Vertical CoM trajectory of the 3-D LIPM generated by an Evolutionary Optimized Central Pattern Generator was proposed in [10]. Guo et al. [11] proposed a modified gait generation method based on CoM height with consine function compensation, but they didn't consider parameters optimization.

Many researchers optimize the gait through the evolutionary algorithm to generate a robust and fast walking. It is a good choice to optimize the gait parameters with machine learning algorithm. It not only provides a great opportunity to experiment on simulation humaniod robots without investing in robot hardware, but also saves a lot of debugging time. Shafii et al.[12] applied Particle Swarm Optimization (PSO) to optimize angular trajectories generated by modified truncated fourier series. MacAlpine successfully used the Covariance Matrix Adaption Evolution Strategy (CMA-ES) algorithm to optimize a large number of gait parameters through overlapping Layered learning for a simulated soccer-playing Humanoid robot [13, 14].

POS was firstly developed by social psychologist Kennedy and electrical engineer Eberhart in 1995 [15]. It is simple and effective to solve global optimization problems, it has been one of the most popular optimization algorithms. POS and its variants have been successfully applied in various fields of science and engineering [16–19]. In order to overcome the shortcoming of premature convergence and improve PSO's performance on complex multimodal problems, Liang et al. proposed comprehensive learning particle swarm optimizer (CLPSO) [20]. CLPSO was considered to have a better performance than other PSO variants [21]. Gülcü Åđ and Kodaz H improved the performance of CLPSO by means of parallel computing and proposed the parallel comprehensive learning particle swarm optimizer(PCLPSO) [21].

In this paper, the sagittal and lateral CoM trajectories are built utilizing numerical method [22] based on IPM [23] to control inverted pendulum dynamics. For the purpose of obtaining a fast, robust and humanlike walking, CoM trajectory in Z axis is not set as a constant value, but generated using quartic polynomial functions. The key parameters need to be optimized are selected from the gait pattern. Then, to shorten the time of gait optimization and improve the performance of biped walking, a parallel distributed training environment is established, distributed gait optimization method is implemented for multiple robots based on modified PCLPSO. RoboCup 3D simulation competitive platform is a distributed multi-robot soccer competition environment [24]. It is easy and suitable to apply PCLPSO to this platform for gait optimization of humanoid robots. We extensively validate and evaluate our algorithm on RoboCup 3D simulation competetitive platform. The experimental and competitive results demonstrate the effectiveness of the method.

2 BIPEDAL WALKING GENERATION

2.1 Vertical CoM Height and ZMP Trajectory Generation

The vertical CoM trajectory could be generated by many methods, such as evolutionary optimized central pattern generator (CPG) [10], or Fourier basis functions [12, 25, 26]. An up-and-down CoM motion is beneficial to fast biped walking [10, 12, 25, 26] . In this paper, vertical CoM trajectory is represented by quartic polynomial function. The equation is given in Eq. (1).

$$c_z(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 \tag{1}$$

Where a_0, a_1, a_2, a_3, a_4, are gait parameters that need to be optimized. Erbatur and Kurt [27] studied the effects of fix and moving ZMP reference trajectories on biped walking, and the research shows that the ZMP reference moving forward under the supporting foot is naturalness and energy efficient. In order to obtain such natural ZMP reference, during the single support phase linear equation is adopted to represent ZMP reference. We assume that the center of supporting foot is origin of coordinate. The sagittal ZMP reference is given in Eq. (2). The lateral ZMP reference is set as the center of supporting foot.

$$p_x(t) = b_1 t + b_0 \tag{2}$$

Where $t \in [0, T_s]$, $b_0 = p_{xmin}$, $b_1 = (p_{xmax} - p_{xmin})/T_s$, p_{xmin} is heel position, p_{xmax} is symbolizes toe position, T_s is step period.

2.2 Horizontal CoM trajectory

IPM is a simple physical model, it assumes that all masses are concentrated on the trunk of the robot and the support leg has no mass. Unlike 3D-LIPM where the vertical CoM height is fixed, the vertical CoM height in IPM can be variable. IPM is well suited for applications of biped walking where the vertical CoM height is variable. We assume that $c = (c_x, c_y, c_z)$ is the CoM position while $p = (p_x, p_y, 0)$ represents ZMP position. if the ZMP trajectory keeps firmly inside the area covered by the support foot, the given biped locomotion will be physically stable and the robot will not fall over while walking. For the purpose of simple bipedal walking planning, angular momentum rate changes are omitted. Hence the ZMP trajectory is respectively planed in the sagittal and lateral plane. The ZMP equations take the form:

$$p_x(t) = c_x(t) - \frac{c_z(t)\ddot{c_x}(t)}{g + \ddot{c_z}(t)} \tag{3}$$

$$p_y(t) = c_y(t) - \frac{c_z(t)\ddot{c_y}(t)}{g + \ddot{c_z}(t)} \tag{4}$$

Where g is the gravitational acceleration.

In Eq. (1) and Eq. (2), vertical CoM height and ZMP trajectory generation have been given. According to Eq. (3) and Eq. (4), horizontal CoM trajectories can be obtained through solving differential equation. Kagami et al. present an effective approach to solve differential equation [22], the ZMP trajectory is discretized with a small time-step Δt by approximate acceleration.

$$i\Delta t \to i, \quad c_x(i\Delta t) \to c_x(i) \tag{5}$$

$$\ddot{c_x}(i\Delta t) \to \frac{c_x(i-1) - 2c_x(i) + c_x(i+1)}{\Delta t^2} \tag{6}$$

Where $i = 1, \cdots, n$. Discrete ZMP equation take the forms:

$$p_x(i) = \alpha_i c_x(i-1) + \beta_i c_x(i) + \alpha_i c_x(i+1) \tag{7}$$

$$\alpha_i = -\frac{1}{\Delta t^2} \frac{c_z(i\Delta t)}{g + \ddot{c_z}(i\Delta t)} \tag{8}$$

$$\beta_i = 1 + \frac{2}{\Delta t^2} \frac{c_z(i\Delta t)}{g + \ddot{c_z}(i\Delta t)} \tag{9}$$

We assume that $c_x(0) = c_x(1), c_x(n) = c_x(n+1)$, then

$$p_x(1) = (\alpha_1 + \beta_1)c_x(1) + \alpha_1 c_x(2) \tag{10}$$

$$p_x(n) = \alpha_n c_x(n-1) + (\alpha_n + \beta_n)c_x(n) \tag{11}$$

Eq. (7) can be expressed in a matrix form as follows:

$$P_x = KC_x \tag{12}$$

Where

$$K = \begin{pmatrix} \alpha_1+\beta_1 & \alpha_1 & 0 & & & & \\ \alpha_2 & \beta_2 & \alpha_2 & & & & \\ & \alpha_3 & \beta_3 & \alpha_3 & & & \\ & & \cdot & \cdot & \cdot & & \\ & & & \cdot & \cdot & \cdot & \\ & & & & \alpha_{n-1} & \beta_{n-1} & \alpha_{n-1} \\ & & & & 0 & \alpha_n & \alpha_n+\beta_n \end{pmatrix},$$

$$P_x = \begin{pmatrix} p_x(1) \\ p_x(2) \\ p_x(3) \\ \cdot \\ \cdot \\ \cdot \\ p_x(n-1) \\ p_x(n) \end{pmatrix}, C_x = \begin{pmatrix} c_x(1) \\ c_x(2) \\ c_x(3) \\ \cdot \\ \cdot \\ \cdot \\ c_x(n-1) \\ c_x(n) \end{pmatrix}.$$

P_x is given by Eq. (2) and matrix K is calculated by Eq. (8), Eq. (9). Then according to Thomas Algorithm, C_x can be obtained by solving the tridiagonal equation Eq. (12). C_x represents the CoM trajectory in sagittal plane. The CoM trajectory in lateral plane can be obtained in the same way.

2.3 Swing Leg Trajectory Generation

The trajectory of swing leg can be generated by sine, cosine, and Bezier curve. The bipedal walking is divided into two stages: the single support phase and the double support phase. If at the start and landing time, the speed and acceleration of the swing leg is zero, it ensures that the biped walking will be stable during the switch between the single support phase and the double support phase. So the trajectory of swing leg is generated based on synthesis of simple harmonic motion in this paper.

$$z_{sw}(t) = \begin{cases} H_{sw}(\frac{2}{T_s}t - \frac{1}{2\pi}sin(\frac{4\pi}{T_s}t)) & t \in [0, \frac{T_s}{2}] \\ H_{sw}(-\frac{2}{T_s}t + \frac{1}{2\pi}sin(\frac{4\pi}{T_s}t) + 2) & t \in (\frac{T_s}{2}, T_s] \end{cases} \quad (13)$$

$$x_{sw}(t) = D_{sw}(\frac{t}{T_s} - \frac{1}{2\pi}sin(\frac{2\pi}{T_s}t)) \quad (14)$$

Where $z_{sw}(t)$ is the vertical trajectory of swing leg , $x_{sw}(t)$ is the trajectory of swing leg in sagial plane, H_{sw} is the maximum height of swing leg foot, D_{sw} is step length.

3 GAIT OPTIMIZATION BASED ON PARALLEL MULTI-SWARM PARTICLE SWARM ALGORITHM

3.1 PSO, CLPSO and PCLPSO

PSO is an efficient optimization approach with stochastic and parallel characteristics. Multiple particles are searching for object in multi-dimensional search space [28]. Compared to other intelligence optimization algorithms, PSO has faster convergence speed and more robust performance. In each iteration, the fitness value of each particle is calculated based on the fitness function. After each iteration, each particle is updated by two best values. The first one is the best fitness it has achieved so far and called "pbest". Another value is global optimum in the whole population which called "gbest" . Hence, The velocity and position of particles in each dimension are updated according to Eq. (15) and (16).

$$\begin{aligned} v_i^d(k+1) &= w * v_i^d(k) + c_1 * rand1_i^d * (pbest_i^d - x_i^d(k)) \\ &+ c_2 * rand2_i^d * (gbest^d - x_i^d(k)) \end{aligned} \quad (15)$$

$$x_i^d(k+1) = v_i^d(k) + x_i^d(k) \quad (16)$$

Where w is inertia weight, $v_i^d(k)$ is the dth component of the velocity vector of the ith particle in the kth iteration, $x_i^d(k)$ is the dth component of the postion vector of the ith particle in the kth iteration. c_1, c_2 are learning factors. $rand1_i^d$ and $rand2_i^d$ are two random numbers between [0,1]. It means that different component in the velocity vector of a particle has different random numbers when updating the particle velocity by Eq. (15).

CLPSO is an effective variant of PSO. A comprehensive learning strategy is proposed in CLPSO. In its velocity updating eqution, a randomly selected *pbest* of all particles is adopted, but the *gbest* is not used. The formula is as follows:

$$v_i^d(k+1) = w * v_i^d(k) + c * rand_i^d * (pbest_{f_i(d)}^d - x_i^d(k)) \quad (17)$$

Where $f_i(d) = [f_i(1), f_i(2), ..., f_i(D)]$ is a randomly selected particle i in the swarm which decided by probability P_c. P_c is gived in Eq. 18. ps is the population size. The detailed flowchart of computing $f_i(d)$ is shown in [20].

$$P_c = 0.05 + 0.45 * \frac{exp(\frac{10(i-1))}{ps-1}))}{exp(10) - 1)} \quad (18)$$

Another feature of CLPSO is the refreshing gap parameter m. This parameter is a switch used for escaping from a local optimum. If a particle can not find a better *pbest* after continuous m iterations, the traditional velocity updating equation including *gbest* will be used once.

PCLPSO is a better variation of CLPSO. The population of particles is divided in multiple swarms in PCLPSO which includes a master-swarm and several slave-swarms. All swarms run PCLPSO in a cluster environment at the same time. The "gbest" ,"lbest", "pbest" in PCLPSO are defined as follows: "gbest" is the global best solution in the population including all swarms, "lbest" is the local best solution of a swarm, "pbest" is the best solution of a particle. After a certain number of iterations, each slave-swarm sends its local best ”*lbest*” of its swarm to the master-swarm. The master-swarm selects the best solution as "gbest" from all local best ”*lbest*” including its own, and then sends the "gbest" to all slave-swarms. Each swarm randomly chooses a particle and this particle updates its ”*pbest*” with the received best solution "gbest" . The distributed parallel computing strategy of PCLPSO contributes to improve the quality of solution and comutational efficiency, shorten the optimize time and enhance the robustness.

3.2 The choice of gait parameter and fitness function

3.2.1 The choice of gait parameters. There are more than 40 gait parameters that need to be optimized in the biped gait optimization problem. If these parameters are optimized altogether, the speed of convergence is influenced because of the enormous searching

Table 1: Gait parameters need to be optimized

Parameter	Description
a_0, a_1, a_2, a_3, a_4	Vertical CoM trajectory
D_s	Step length
D_w	Step width
H_{sw}	Maximum height of the swing leg foot
S_θ	Turn body angle
T_s	Step period
T_d	Period for the double support foot
Zmp_{off}	Constant offset of ZMP
x_f	Distance from the center of two feet to CoM

space. Hence, in this paper we just chose several key parameters, as shown in table 1, which affect the walking speed and stability for the gait optimization.

3.2.2 fitness function. A good evaluation criterion plays an very important role in optimization results. The following evaluation criteria were used for judging a gait.

1) Walking distance during one period.

$$f_{dis} = \|\phi_{end} - \phi_{start}\| \tag{19}$$

Where ϕ_{start} is the initial coordinate of robot during training, ϕ_{end} is the last coordinate of robot at the end of a training period.

2) As is shown in [29], if the ZMP trajectory is kept firmly inside the support polygon, the robot will not fall down while walking. In this paper, ZMP coordinate of walking sequence will be considered to evaluate the gait.

$$f_{zmp} = \sum_{k=1}^{N} \sqrt{(\hat{p}_x(k) - p_x(k))^2 + (\hat{p}_y(k) - p_y(k))^2} \tag{20}$$

Where $p_x(k)$, $p_y(k)$ is the expected ZMP coordinate of walking sequence, $\hat{p}_x(k)$, $\hat{p}_y(k)$ is the measured ZMP coordinate of walking sequence.

3) The trunk of the robot should be always stable and do not shake while walking. During training, at the initial time of the double support phase, the expected CoM coordinate is the center of two feet. But in fact there is always a deviation between two coordinates. So the trunk shaking is detected by comparing two coordinates between CoM and the center of two feet at the initial time of the double support phase.

$$x_f = c_x - \frac{x_{footR} + x_{footL}}{2} \tag{21}$$

$$f_{shake} = \begin{cases} 0 & fabs(x_f) < th_\theta \\ c & otherwise \end{cases} \tag{22}$$

x_{footR}, x_{footL} are the coordinates of two feet at the initial time during the double support phase. th_θ is the set threshold value, f_{shake} is penalty value and c is a positive constant.

If the robot falls down during training, a positive constant $f_{falling}$ will be given as a penalty value. Hence, the fitness function is as follows

$$F = \gamma_1 f_{dis} - \gamma_2 f_{zmp} - \gamma_3 f_{shake} - f_{falling} \tag{23}$$

Where $\gamma_1, \gamma_2, \gamma_3$ are discount factors.

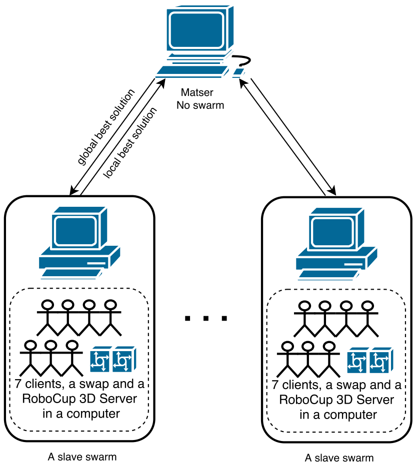
3.3 Gait optimization based on PCLPSO in distributed environment

In the RoboCup 3D simulation competition, a client is a independent program which control a simulated NAO robot. All clients connect to the RoboCup 3d simulated soccer server. The clients and the server can run on multiple computers in a distributed environment. The cluster structure based on PCLPSO and RoboCup 3D simulated environment is shown in Fig. 1(a), which a swarm runs on a computer, Fig. 1(b) shows another structure which a slave swarm runs on multiple computers.In [21], all particles in a swarm run on the same computer in PCLPSO. So it is not suitable to apply PCLPSO directly to the RoboCup 3D simulated environment for optimization .

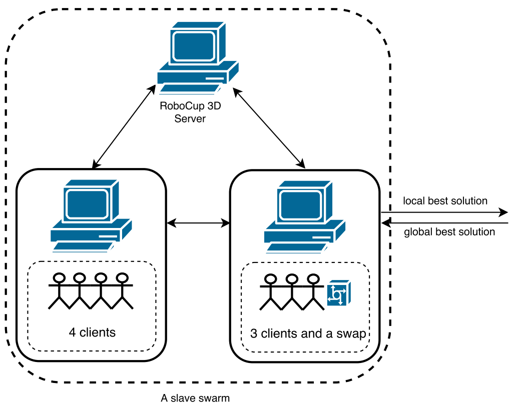
In this paper, PCLPSO is decomposed into three algorithms to adapt to our new cluster structure. In our cluster structure, the master has no swarm, and its main function is to send initial parameters to slave swarms, collect "lbest" from slave swarms and send "gbest" to slave swarms. The detailed algorithm of the master is shown in Alg. 1. A swap program is added in slave swarm for the purpose of swapping data between the master and all clients of a slave swarm, as is shown in Alg. 2. A client is a particle of a swarm. This is the main section of gait optimization based on PCLPSO, including calculating ZMP Trajectory, 3D CoM Trajectory and Swing leg Trajectory based on bipedal walking generation, calculating all joint angles of Robot NAO in a walk cycle, performing a walk in Robocup 3D simulation environment calculate fitness and updating the velocity and position based on PCLPSO. The detailed algorithm of a client is shown in Alg. 3. Gait optimization process of a client based on PCLPSO is shown in Fig. 2.

Algorithm 1 The PCLPSO Algorithm in Master

int n /* Number of the swarms */
int k /* Population size of each swarm */
int P /* Migration period */
Array EP /* ElitePool, it stores the local best solutions */
int m /* the refreshing gap */
double $w, c1, c2$ /* the inertia weight and the acceleration coefficients */
Initialize the parameters $n, k, P, m, w, c1, c2$
Wait until all slave swarms have been launched and connected to this Master
The master sends the parameters to the slaves
repeat
  Wait until all slave swarms have sent lBest to the master
  The master stores the lBests into the EP
  The master finds the gBest in the EP and empties the EP
  The master sends the gBest to the slaves
until the stopping criterion is met
return



(a) The cluster structure in which a swarm runs on a computer



(b) A slave swarm runs on multiple computers

Figure 1: The cluster structure based on PCLPSO and RoboCup 3D simulated environment

Algorithm 2 The PCLPSO Algorithm in the swap program of a slave swarm

```
int n,k,P,m
double w, c1, c2
Array LP          /* LocalPool, it stores the all pbests in a swarm */
Connect to the Master
Wait until the parameters are received from the master
Send the parameters to the clients in this swarm
int d = 0
repeat
    d + +
    Wait until all pBests in this slave swarm are received
    Store the pBests into the LP
    if d mod P == 0 then
        Find the lBest in the LP
        Send the lBest to the master
        Wait until the gBest is received from the master
        Randomly update a lBest with the gBest in the LP
    end if
    Send the LP to the all clients in this swarm
until the stopping criterion is met
return
```

4 EXPERIMENTS AND RESULT

The work is fully implemented and validated using simulation Nao which is currently used in RoboCup 3D simulated platform . Biped robot NAO has 57cm and 22 degrees of freedom(DOF). RoboCup 3D Server uses the Open Dynamics Engine(ODE) for its realistic simulation of rigid body dynamics with collision detection and friction [24]. Gait optimized in simulation can also translate the application in real Aldebaran Nao [26]. A training scene graph of a slave swarm is shown in Fig. 3.

4.1 Experiment 1: contrast between Multi-swarm PSO and PSO

In order to verify the effectiveness of the proposed distributed algorithm based on the parallel multi-swarm PSO, we firstly optimize the forward walk based on the evaluation criterion of speed and stability. Eq. 23 is used as the fitness function. The trajectories of best fitness value with constant CoM height and ZMP trajectory are shown in Fig. 4. Fig. 4 shows that the best fitness value increases with increasing the number of iterations. When gait parameters are optimized based on multi-swarm PSO, after approximately 80 iterationsthe the best fitness value reaches the maximum and the biped robot which used these optimized gait parameters can walk 6.5m in 10 seconds with the average speed of 0.65m/s. While the ultimate average speed is 0.56m/s when the gait parameters is optimized based on PSO. Compared to PSO, the effectiveness of gait optimization based on multi-swarm PSO is also reflected in the walking stability of the biped robot. Walking stability detected by Eq. 21 is shown in Fig. 5. The x_f based On multi-swarm PSO is in a small range compared to PSO. The optimized trajectories of swing leg for forward walking is shown in Fig. 6 and Fig. 7. At the start and landing time, the optimized trajectory of swing leg is parallel to the ground which ensure the stability during the switch between the single support phase and the double support step phase.

4.2 Experiment 2: contrast between variable and constant trajectories of CoM height and ZMP

To verify the effectiveness of the proposed trajectories generation of vertical CoM height and ZMP, the parameters of Eq. 1 and Eq. 2 are optimized. Before training , the parameters of the gait are set based on previous experience of debugging and the biped gait is slow and has poor stability. We optimize the forward walk with variable vertical CoM height based on multi-swarm PSO in the same way as experiment 1. the biped robot can walk 7.4 m in 10 seconds with the average speed of 0.74m/s. The result shows that the speed of biped robot with variable vertical CoM height and ZMP trajectories is faster than the speed of biped robot with fixed vertical CoM height and ZMP trajectories. Fig. 8 shows optimized vertical CoM

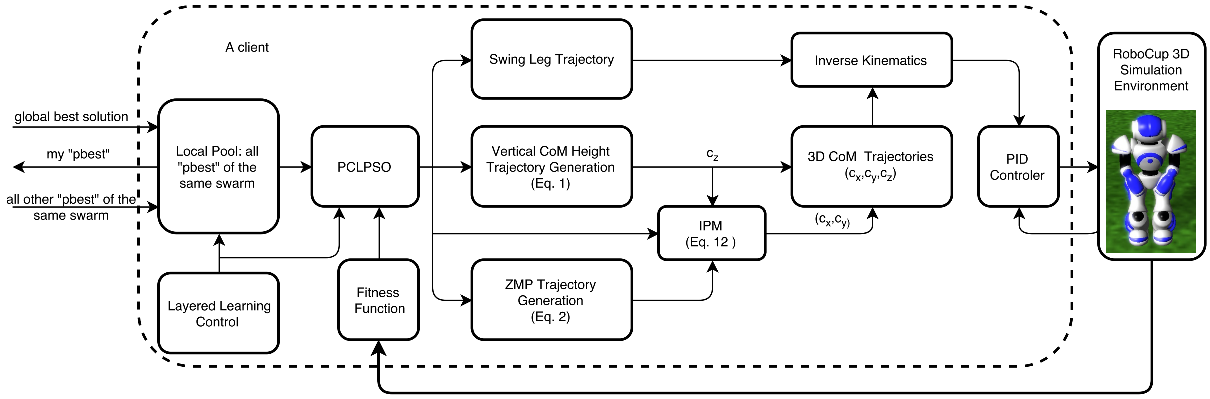


Figure 2: Gait optimization process based on PCLPSO.

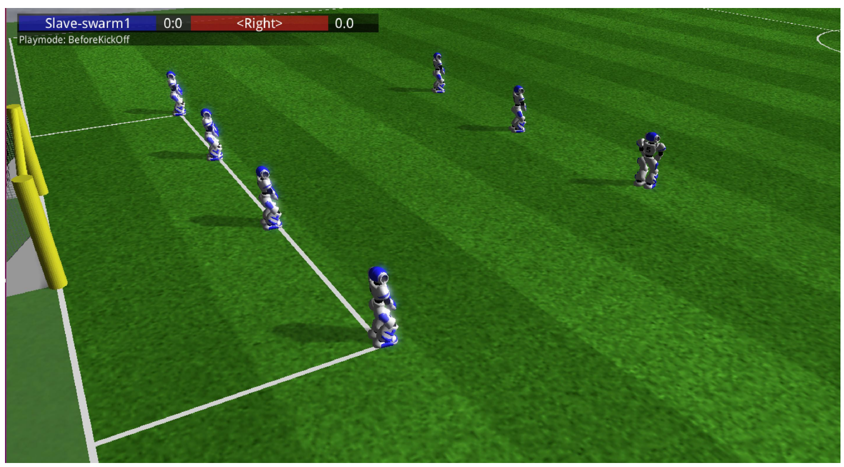


Figure 3: Training scene of a slave swarm

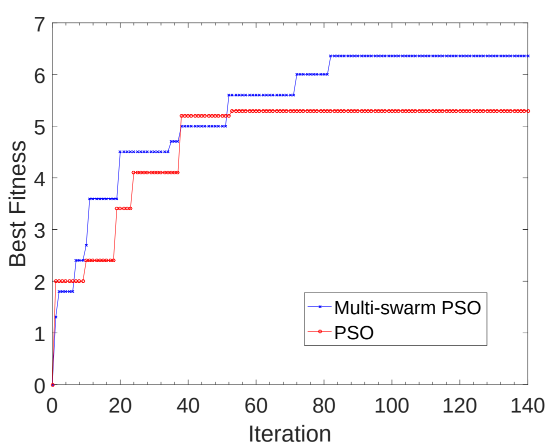


Figure 4: Converge trajectories of fitness function

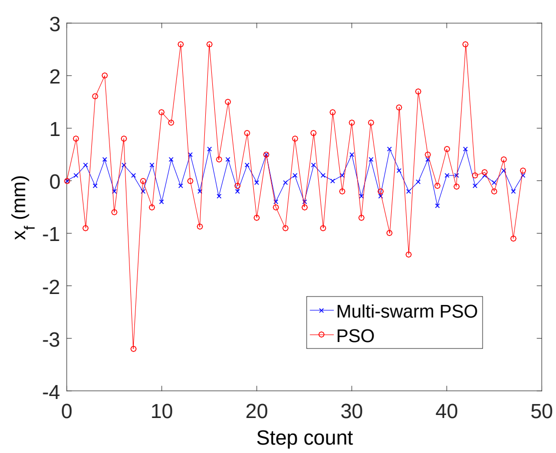


Figure 5: Walking stability detected by Eq. 21

trajectory. Fig. 9 shows optimized CoM and ZMP trajectory in x direction. Gait parameters which have been optimized for forward walk is shown in Table 2.

Algorithm 3 The PCLPSO Algorithm in a client of a slave swarm

int n,m
double w, $c1$, $c2$
Array LP /* LocalPool, it stores the all pbests in a swarm */
Connect to the swap program in a swarm
Wait until the Parameters are received from the swap program in a swarm
Initialize the Parameters n, m, w, $c1$, $c2$
Initialize position and velocity
Calculate ZMP Trajectory, 3D CoM Trajectory and Swing leg Trajectory based on bipedal walking generation
Calculate all joint angles of Robot NAO in a walk cycle
Perform a walk in RoboCup 3D simulation environment
Calculate fitness
Update pBest /* Update the pBest of the particle */
Send the pBest to the swap program
Wait until the LP is received from the swap program
Update the LP
Find the lBest in the LP
repeat
    Update the velocity and position based on PCLPSO
    Calculate ZMP Trajectory, 3D CoM Trajectory and Swing leg Trajectory based on bipedal walking generation
    Calculate all joint angles of Robot NAO in a walk cycle
    Perform a walk in Robocup 3D simulation environment
    Calculate fitness
    Update pBest /* Update the pBest of the particle */
    Send the pBest to the swap program
    Wait until the LP is received from the swap program
    Update the LP
    Find the lBest in the LP
until the stopping criterion is met
return

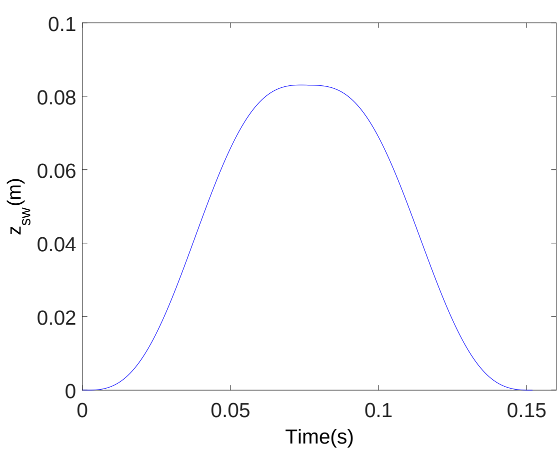


Figure 6: Optimized trajectory of swing leg in z direction

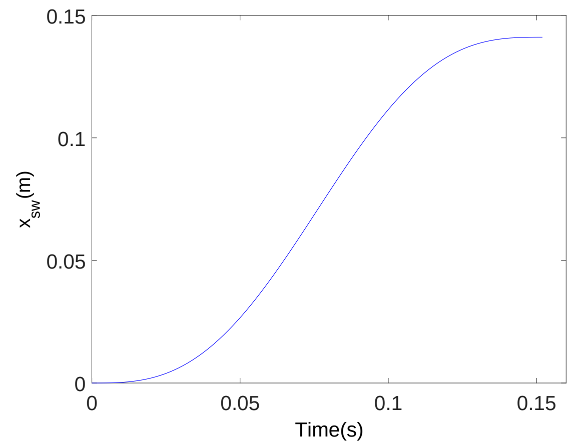


Figure 7: Optimized trajectory of swing leg in x direction

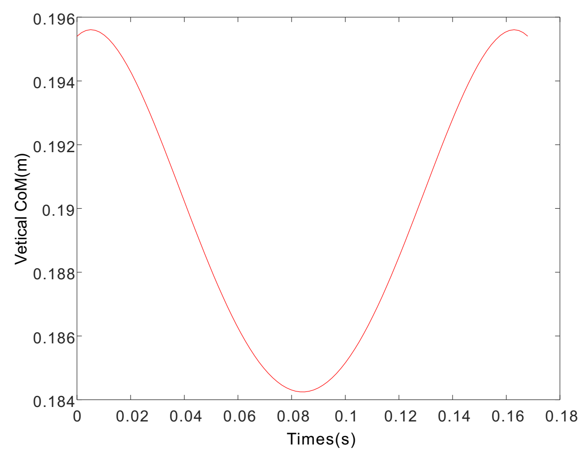


Figure 8: Optimized vertical CoM trajectory

Table 2: Gait parameters have been optimized

Parameter	Optimized value
a_0	0.1954
a_1	0.0831
a_2	-8.7899
a_3	98.7102
a_4	-293.648
D_s	0.141m
H_{sw}	0.083m
T_s	0.168s
T_d	0.016s

4.3 Experiment 3: the layered learning of gait optimization

In order to adapt to the dynamic walk of biped robot, the gait should to be further optimized. In this paper, the layered learning [14, 30] of gait optimization is applied. The parameters selected to

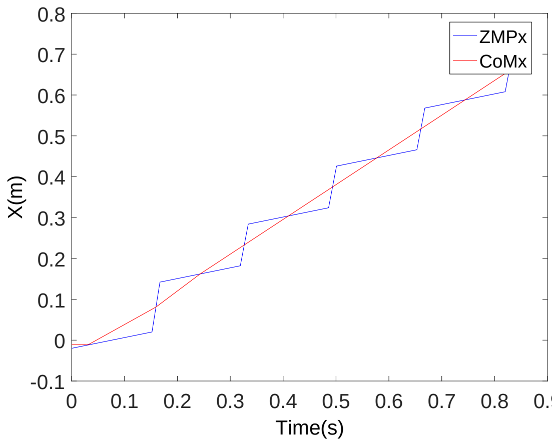


Figure 9: Optimized CoM and ZMP trajectory in x direction

Table 3: Turning performance test

Walking distance (m)	Initial angle (°)	Time (s)	Optimized speed (m/s)
15	30	21.27	0.705
15	45	21.45	0.699
15	60	21.78	0.689
15	90	22.42	0.669

be optimized and the range of these parameters are added step by step in the gait Optimization. For training an omnidirectional walk, the gait parameters of lateral walking and the turning angle are added in the optimization based on experiment 2. The maximum distance of lateral walking is 0.04 and the maximum turning angle is 15Âřin each step. The two different target points are set up for the biped robot walking in the optimization process. Table 3 shows the results of turning performance test. The initial angle is the angle between the body orientation of robot and the target point at the training initial time. The results in table 3 show that the layered learning of gait optimization is effective.

5 CONCLUSIONS

In this paper, distributed gait optimization method is presented for multiple robot based on multi-swarm PSO. The experimental results demonstrate that the proposed algorithm shows better performance than traditional PSO. The proposed algorithm could help to greatly shorten the time of gait optimization and improve the efficiency of gait optimization. Meanwhile, a machine learning method of optimizing the vertical CoM trajectory is presented. Simulation experiments show that the biped robot with variable vertical CoM height and ZMP trajectory could walk faster than the biped robot with fixed vertical CoM height and ZMP trajectory. Finally, the gait optimization method based on layered learning is presented to achieve an omnidirectional walk for a biped robot. The gait optimized by the presented method has been successfully realized in our CIT3D team. And our CIT3D team won twice second runner-up in the RoboCup ChinaOpen(2016, 2017). Compared to CMA-ES used in [13, 14, 26], we proposed another effective method for gait optimization. Cloud training environment based on the proposed algorithm will be established to train all kinds of skills for biped robot in the future.

ACKNOWLEDGMENTS

This work is supported by the Natural Science Foundation of the Jiangsu Higher Education Institutions of China (Grant No.: 17KJD520001, Grant No.: 16KJB520003, Grant No.: 17KJB120011) and the Natural Science Foundation of Changzhou Institute of Technology (Grant No.: YN1603).

REFERENCES

[1] Grizzle J W, Chevallereau C, Sinnet R W, et al. Models, feedback control, and open problems of 3D bipedal robotic walking[J]. Automatica, 2014(50):1955-1988.
[2] www.robocup.org
[3] Kajita S, Kanehiro F, Kaneko K, et al. A realtime pattern generator for biped walking[C]//Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on. IEEE, 2002, 1: 31-37.
[4] Kajita S, Kanehiro F, Kaneko K, et al. Biped walking pattern generation by using preview control of zero-moment point[C]//Proceedings of IEEE International Conference on Robotics and Automation. IEEE, 2003, 2: 1620-1626.
[5] Graf C, RÃűfer T. A center of mass observing 3D-LIPM gait for the RoboCup Standard Platform League humanoid[C]//Springer Berlin Heidelberg:Robot Soccer World Cup , 2011: 102-113.
[6] Guo F, Mei T, et al. A generic walking pattern generation method for humanoid robot walking on the slopes[J]. Industrial Robot: An International Journal, 2016, 43(3): 317-327.
[7] Jadidi M G, Hashemi E. Optimal preview control of the Nao biped robot using a UKF-based state observer[C]//Advanced Intelligent Mechatronics (AIM), 2016 IEEE International Conference on. IEEE, 2016: 52-57.
[8] Zhang L, Zhou C. Optimal three-dimensional biped walking pattern generation based on geodesics[J]. International Journal of Advanced Robotic Systems, 2017, 14(2): dx.doi.org/10.1177/1729881417696235.
[9] Kuo A D, Donelan J M, Ruina A. Energetic consequences of walking like an inverted pendulum: step-to-step transitions[J]. Exercise and sport sciences reviews, 2005, 33(2): 88-97.
[10] Hong Y D, Lee K B. Stable Walking of Humanoid Robots Using Vertical Center of Mass and Foot Motions by an Evolutionary Optimized Central Pattern Generator[J].International Journal of Advanced Robotic System, 2016.13:27.
[11] Guo F Y, Zhao J H, et al. A modified gait generator for humanoid robots based on height compensation of center of mass[C]//Robotics and Biomimetics (ROBIO), 2014 IEEE International Conference on: IEEE, 2014: 1278-1283.
[12] Shafii N, Aslani S, Nezami O M, et al. Evolution of biped walking using truncated fourier series and particle swarm optimization[C]//Robot Soccer World Cup: Springer Berlin Heidelberg, 2009: 344-354.
[13] MacAlpine P, Barrett S, Urieli D, et al. Design and Optimization of an Omnidirectional Humanoid Walk: A Winning Approach at the RoboCup 2011 3D Simulation Competition[C]//In the Twenty-Sixth Conference on Artificial IntelligenceïijĹAAAI-12ïijĽ,Toronto,Ontario, Canda, 2012:1047-1053.
[14] MacAlpine P, Depinet M, Stone P. UT Austin Villa 2014: RoboCup 3D Simulation League Champion via Overlapping Layered Learning[C]//in the Twenty-Ninth AAAI Conference Intelligence(AAAI-15).Austin,texas,USA: 2015: 2842-2848.
[15] Kennedy J, Eberhart R C. (1995). Particle swarm optimization. In Proceedings of the IEEE international conference on neural networks IV (pp. 1942âĂŞ1948). Piscataway: IEEE.
[16] Poli R, Kennedy J, Blackwell T. Particle swarm optimization: An Overview[J]. Swarm intelligence, 2007, 1(1): 33-57
[17] Bonyadi M R, Michalewicz Z. Particle swarm optimization for single objective continuous space problems: a review[J].Evolutionary Computation,2017, 25(1): 1âĂŞ54.
[18] Esmin A A A, Coelho R A, Matwin S. A review on particle swarm optimization algorithm and its variants to clustering high-dimensional data[J]. Artificial Intelligence Review, 2015, 44(1): 23-45.
[19] Jordehi A R, Jasni J. Particle swarm optimisation for discrete optimisation problems: a review[J]. Artificial Intelligence Review, 2015, 43(2): 243-258.
[20] Liang J J, Qin A K, Suganthan P N, et al. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions[J]. IEEE transactions on evolutionary computation, 2006, 10(3): 281-295.
[21] Gülcü Åđ, Kodaz H. A novel parallel multi-swarm algorithm based on comprehensive learning particle swarm optimization[J]. Engineering Applications of Artificial Intelligence, 2015, 45: 33-45.

[22] Kagami S, Kitagawa T, Nishiwaki K, et al. A fast dynamically equilibrated walking trajectory generation method of humanoid robot[J]. Autonomous Robots, 2002, 12(1): 71-82.
[23] Kajita S, Nagasaki T, Yokoi K, et al. Running pattern generation for a humanoid robot[C]//ICRA. 2002: 2755-2761.
[24] Xu Y, Vatankhah H. Simspark: an open source robot simulator developed by the RoboCup community[C]//Robot Soccer World Cup. Springer Berlin Heidelberg, 2013: 632-639.
[25] Shafii N, Lau N, Reis L P. Learning a fast walk based on ZMP control and hip height movement[C]// Autonomous Robot Systems and Competitions (ICARSC), 2014 IEEE International Conference on IEEE, 2014:181-186.
[26] Shafii N, Lau N, Reis L P. Learning to Walk Fast: Optimized Hip Height Movement for Simulated and Real Humanoid Robots[J]. Journal of Intelligent & Robotic Systems, 2015, 80(3-4): 555-571.
[27] Erbatur K, Kurt O. Natural ZMP trajectories for biped robot reference generation[J]. IEEE Transactions on Industrial Electronics, 2009, 56(3): 835-845.
[28] Shi YïijŇEbemart R C. A modified particle swam optimizer[C]// Proceedings of 1998 IEEE International Conference on Evolutionary Computation. New York, NY, USA: IEEE, 1998: 69-73.
[29] Vukobratović M, Borovac B. Zero-moment pointâĂŤthirty five years of its life[J]. International Journal of Humanoid Robotics, 2004, 1(01): 157-173.
[30] Chunguang Li. Research on Motion Planning and Flocking Control For Humanoid Robot[D]. Wuxi: Jiangnan University, 2015.