Multi-Objective Optimization in Gait Planning of Biped Robot Using Genetic Algorithm and Particle Swarm Optimization Algorithm

Rajendra Rega, Dilip Kumar Pratihar*
Soft Computing Laboratory, Mechanical Engineering Department
Indian Institute of Technology, Kharagpur, India
*dkpra@mech.iitkgp.ernet.in

***Abstract*—This paper deals with multi-objective optimization in gait planning of a 7-dof biped robot ascending and descending some staircases. Both its power consumption as well as dynamic balance margin depends on a few common design parameters. The biped robot should have a maximum dynamic balance margin but at the expense of minimum power. Thus, a conflicting relationship exists between these two objectives. The said gait planning problem has been modeled and solved using two modules of adaptive neuro-fuzzy inference system. The said multi-objective optimization problems have been solved using a genetic algorithm and particle swarm optimization algorithm, separately. Pareto-optimal fronts of solutions have been obtained, which may help a designer to select the most appropriate solution out of several possibilities. Particle swarm optimization algorithm is found to perform better than genetic algorithm, as the former performs both local and global searches simultaneously, whereas the latter is seen to be weak in terms of its local search capability. Therefore, the main contribution of this paper lies with the application of two optimization algorithms to tackle multi-objective optimization in gait planning of biped robot.**

Keywords—Biped robots; Gait planning; Multi-objective optimization; Pareto-optimal front; Genetic algorithm; Particle swarm optimization.

I. INTRODUCTION

Recent research in robotics aims to build intelligent, energy efficient and dynamically balanced biped robots capable of moving through various terrains, as the situation demands. Intelligence is developed in a robot artificially, in the form of adaptive motion (path and/or gait) planner. The robot can be made energy efficient through the optimization of its mechanical structure. Moreover, Dynamic Balance Margin (DBM) of a biped robot can be measured using the concept of Zero Moment Point (ZMP) [1]. A ZMP is defined as a point lying on the foot-ground contact plane, about which the algebraic sum of all moments becomes equal to zero. In order to maintain the dynamic balance, the ZMP should lie within the support polygon of the biped robot.

A considerable amount of effort was made in the past to design and develop energy efficient biped robots and their gaits using evolutionary algorithms. Some of those studies are discussed here. Capi et al. [2] used a Genetic Algorithm (GA) in order to minimize energy consumption in trajectory generation of a biped robot for its stable walking on flat surface and its performance was verified experimentally. Capi et al. [3] developed another approach using a GA and Radial Basis Function Neural Network (RBFNN) for gait synthesis of a biped robot in terms of two objectives like minimum consumed energy and minimum change in torque. The optimal gaits determined by the GA were used to train the RBFNN for on-line gait generations of the robot, but their approach was found to be computationally expensive. Jeon et al. [4] utilized a real-coded GA to minimize total energy consumption of a biped robot ascending and descending the staircase. The optimal gait trajectory was generated for stable walking on a staircase. Results of Matlab simulations showed that a significantly higher amount of energy is consumed for ascending the staircase compared to that necessary for descending the staircase. Salatian and Zheng [5,6] used a Neural Network (NN) to update rhythmic motion of a two-legged robot walking on slopping surface. The gait parameters were dynamically modified by the network according to the change in slope of the surface. Fan et al. [7] developed an approach for real-time gait generation of a biped robot utilizing a Fuzzy Neural Network (FNN). They used Matlab software to determine optimal gait parameters for a biped robot after considering three energy consumption indices, namely mean power, mean power deviation and mean torque change. The minimum energy gaits were used to train the FNN.

Several attempts were also made by various investigators to maximize dynamic balance margin of biped robots. Some of those studies are discussed here. Kun and Miller III [8] proposed an adaptive dynamic balance scheme for a biped robot using Cerebellar Model Arithmetic Computer (CMAC) NNs, which were responsible for maintaining good balance and foot contact. Miller III [9] developed control strategies based on hierarchy of simple gait oscillators, PID controllers, and NN learning without using the detailed dynamic or kinematics models. Real-time control of a 10-dof biped robot with force sensing capability was implemented, and the experimental biped robot could learn the quasi-static balance required to avoid falling and maintain dynamic balance necessary for lifting the foot. Zhou and Meng [10] developed a neuro-fuzzy network, in which a Fuzzy Reinforcement Learning (FRL) method had been utilized for dynamic control of the biped robot. Jha et al. [11] used a genetic-fuzzy system for on-line gait generation for a biped robot. A GA was utilized to optimize the rule base of Fuzzy Logic Controller (FLC) offline. The optimized FLC was able to develop on-line stable gaits for the biped robot. Udai [12] utilized a GA to optimize hip

trajectory of a biped robot during its single support phase. Here, the objective was to minimize the deviation of ZMP from the geometric centre of supporting foot area using the input parameters like swing foot trajectory and physical parameters of the robot. The results were verified through Matlab simulations. Vundavilli et al. [13] developed two different hybrid approaches, namely GA-NN and GA-FLC systems for dynamically balanced gait generation of a biped robot ascending and descending some staircases. The GA was used to optimize the weights of NN in GA-NN and rule base in GA-FLC systems, offline. The optimized GA-NN and GA-FLC approaches were able to successfully generate dynamically balanced gaits for the biped robot in computer simulations.

The issues related to gait planning of biped robots had been formulated as multi-objective optimization problems also, where each of the objectives was a function of some common design variables. Moreover, those objectives were seen to contradict each other, so that Pareto-optimal front of solutions could be obtained. In these connections, the studies of [14-16] are worth mentioning. Lee and Lee [14] generated walking patterns for the best performance of a biped robot using multi-objective evolutionary algorithm. They considered three contrasting objectives, namely mobility, energy efficiency and stability of a robot to obtain optimal set of solutions for generating walking gaits on flat surface. Capi et al. [15] used a multi-objective evolutionary algorithm to determine optimally stable gaits of a biped robot walking on flat surface, after considering the objectives like minimization of energy consumption and change in torque simultaneously. The results were successfully implemented on BONTEN-MARU biped robot for flat surface walking. Goswami et al. [16] carried out a GA-based optimal bipedal walking gait synthesis considering a trade-off between stability margin and walking speed. The stable gait walking parameters were optimized using a GA, verified through Matlab simulations and then implemented on an experimental biped robot walking along a straight path on flat surface.

Kennedy and Eberhart [17] proposed a concept for the optimization of nonlinear functions using Particle Swarm Optimization (PSO) algorithm. It fulfills the necessary conditions laid in accordance with a paper by Millonas [18], who developed his model for applications in artificial life, and articulated five basic principles of swarm intelligence, namely proximity, quality, diverse response, stability and adaptability. Coello Coello et al. [19] developed a multi-objective PSO (MOPSO), which was able to cover the full Pareto-optimal front of solutions. However, they did not use the concept of crowding distance proposed by Deb et al. [20] in NSGA-II. Reyes-Sierra and Coello Coello [21] predicted the future trends of the current MOPSO. It would be more efficient and self–adaptive in nature. They opined that more theoretical work should be done to make it applicable to real-time industrial problems. Poli [22] reviewed the articles and indicated the broad application areas of PSO. He forecasted its immense scope of applications in various areas like medical science, electrical and electronics science, combinatorial problems, image analysis, signal analysis, graphics, robotics, and others. Sivakumar et al. [23] compared the performances of NSGA-II and MOPSO in terms of the quality of Pareto-optimal front, spread of solutions, strength of non-dominated individuals in the optimal fronts, computational complexity, and others. MOPSO could outperform the NSGA-II with respect to all the above mentioned parameters. Rokbani et al. [24] utilized the PSO to optimize gait stability of the biped robot. The obtained optimal results were used to test the stability of the biped robot. The PSO algorithm could generate optimal angular positions of joints for the stable walking. The obtained optimal results were used to test the stability of a biped robot kit for its standing posture on flat surface. Niehaus and R¨ofer [25] used the PSO algorithm for walking gait optimization of a humanoid robot. The biped gait was modeled utilizing a number of parameterizable trajectories to achieve omni-directional walking. The optimized set of walking parameters was successfully implemented on a modified Kondo KHR-1 robot on flat surface. Kim et al. [34] used nonparametric estimation-based PSO for finding the parameters of Central Pattern Generator (CPG). The PSO algorithm was able to efficiently determine CPG parameters for a biped gait.

Most of the studies on biped robots available in the literature are related to their walking on flat surface. However, a biped robot should be able to walk on rough terrains also, such as staircases, and others. The problems related to locomotion of biped robots on rough terrains are more complex (from the analysis and control points of view) compared to those on flat surface. Rough terrain locomotion of biped robots has not yet received much attention, till date.

In the present work, an attempt has been made to formulate gait planning problem of a 7-dof biped robot ascending and descending the staircase as a multi-objective optimization one. Two modules of adaptive neuro-fuzzy inference system (ANFIS) have been utilized to model gait planning problem of the biped robot. Two conflicting objectives, such as minimization of power consumption and maximization of dynamic balance margin have been considered in the present study, and a GA and a PSO algorithm have been utilized to yield Pareto-optimal front of solutions separately. A comparison on the performances of these two optimization algorithms has also been presented.

The remaining part of this paper has been organized as follows: Section II deals with mathematical formulation of the problem. Section III explains the proposed algorithms to solve the said problems. Results are stated and discussed in section IV. Some concluding remarks are made in section V.

II. MATHEMATICAL FORMULATION OF THE PROBLEM

The present study deals with an analysis of a 7-dof (that is, three at hip, two at knee and two at ankle) biped robot ascending and descending some staircases. It consists of two ankles, two lower legs, two upper legs and a trunk connected through seven rotary joints.

A. Staircase Ascending

Fig.1 displays the schematic view of a biped robot ascending a staircase. The biped walking cycle consists of two phases: single support phase (SSP) (that is, when one foot is in contact with the ground and the other is in the air) and double support phase (DSP) (that is, when both the feet are in contact

with the ground). The present study concentrates on SSP only of a biped robot ascending and descending the staircases. The mass of each link is assumed to be concentrated at a point lying on it. For simplicity, the movement of the biped robot and its balance is considered in one direction only. During motion, the swing foot of the robot is assumed to follow a cubic polynomial trajectory, so as to avoid collision with the staircase, as given below.

$$z = c_0 + c_1 x + c_2 x^{2,} + c_3 x^3 \tag{1}$$

where z represents the height of the swing foot from the surface of the lower staircase, at a distance of x from the starting point, and c_0 , c_1 , c_2 and c_3 are the coefficients, whose appropriate values are to be determined using the boundary conditions given below.

$$\alpha\tau \;\; x = 0, z = 0;$$

$$\alpha\tau \;\; x = s_w - x_1 - f_s/2, z = s_h + f_s/2;$$

$$\alpha\tau \;\; x = 2s_w - x_1, z = 2s_h + f_s/2;$$

$$\alpha\tau \;\; x = 2s_w - x_1 + x_3, z = 2s_h;$$

where s_w and s_h denote the width and height of the staircase, respectively, f_s represents the length of the swing foot. In a cycle, the swing foot is assumed to follow a velocity distribution as shown in Fig. 2.

The hip trajectory is assumed to follow a straight line having a slope equal to that of the staircase to ensure repeatability conditions of the cycles. The angles made by the lower and upper parts of swing leg, that is, θ_2 and θ3 are calculated based on the hip and swing foot trajectories, as given below.

$$\theta_2 = \sin^{-1}\left(\frac{h_1 L_3 \sin\varphi_1 + l_1(L_2 + L_3\cos\varphi_1)}{(L_2 + L_3\cos\varphi_1)^2 + (L_3\sin\varphi_1)^2}\right) \tag{2}$$

where h_1 and l_1 are shown in Fig 1; L2 and L3 are the lengths of two links; and $\varphi_1 = \cos^{-1}((h_1^2 + l_1^2 - L_2^2 - L_3^2)/2L_2L_3)$. Thus, θ_3 can be calculated as $\theta_3 = \theta_2 - \varphi_1$ .Similarly, the angles: θ_5 and θ_6 can be determined using the information of hip height (h2) and distance of the supporting ankle from the projection of hip joint (l2) as given below (refer to Fig. 1).

$$\theta_6 = \sin^{-1}\left(\frac{h_2 L_5 \sin\varphi_2 + l_2(L_6 + L_5\cos\varphi_2)}{(L_6 + L_5\cos\varphi_2)^2 + (L_5\sin\varphi_2)^2}\right) \tag{3}$$

where L5 and L6 are the lengths of the links : $\varphi_2 = \cos^{-1}((h_2^2 + l_2^2 - L_6^2 - L_5^2)/2L_5L_6)$. The angle: θ5 can be obtained using the expression $\theta_5 = \theta_6 - \varphi_2$. A cycle has been divided into seven equal intervals. The following repeatability conditions are maintained to generate a cyclic gait:

$$\theta_{2,initial} = \theta_{6,final};\ \dot{\theta}_{2,initial} = \dot{\theta}_{6,final};\ \theta_{3,initial} = \theta_{5,final};$$
$$\dot{\theta}_{3,initial} = \dot{\theta}_{5,final}.$$

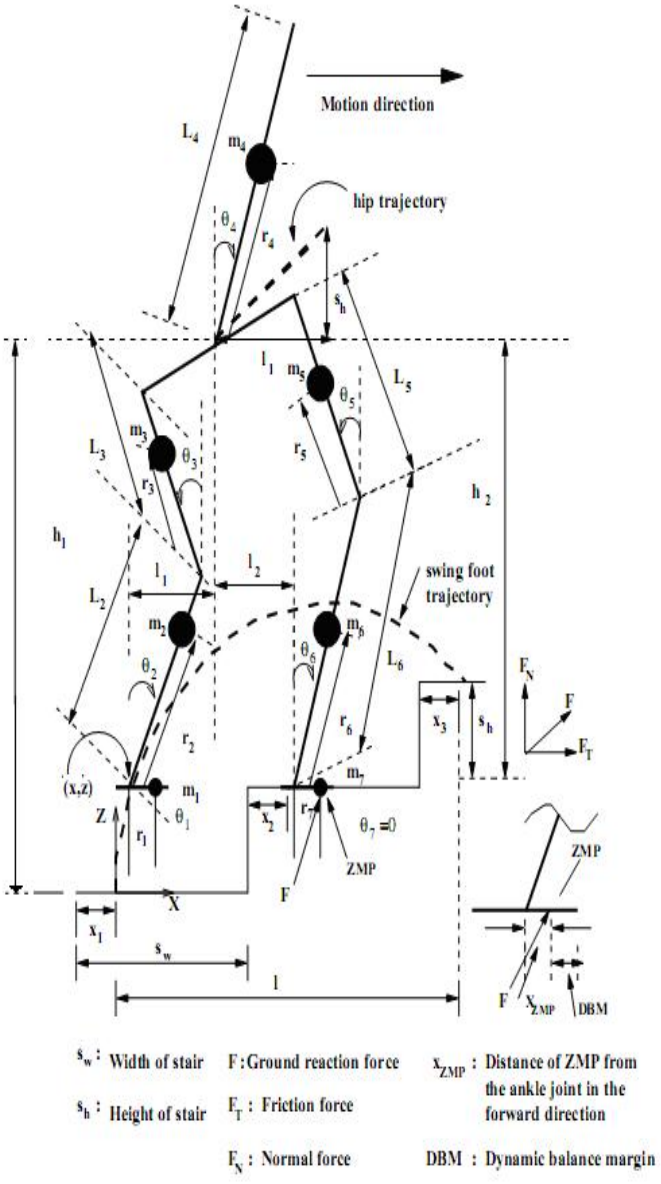


Fig.1 A schematic view of a biped robot ascending the staircase showing the hip joint and swing foot trajectory. (Assumption: three joints coincide at the hip).

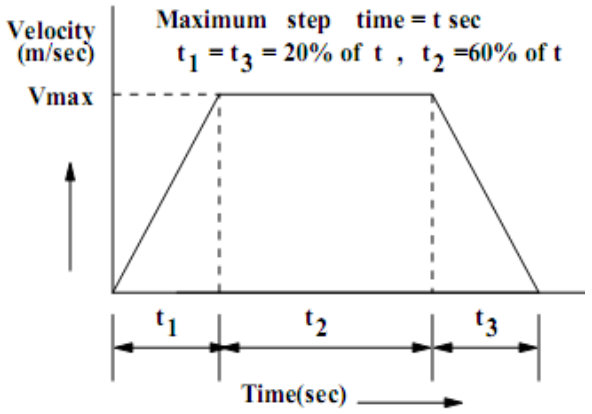


Fig. 2 Velocity profile of the swing foot.

The values of trunk and swing foot angles will be determined using two adaptive neuro-fuzzy inference systems (ANFIS), which will be discussed in the next section. The motions of trunk and swing foot will have to fulfill the repeatability conditions, so that the postures at the beginning and end of the cycle remain the same. In order to maintain repeatability of the trunk motion, $\theta_{4,final}$ and $\dot{\theta}_{4,final}$ are kept equal to $\theta_{4,initial}$ and $\dot{\theta}_{4,initial}$, respectively, and for the swing foot, $\theta_{1,final}$ and $\dot{\theta}_{1,final}$ are made equal to $\theta_{1,initial}$ and $\dot{\theta}_{1,initial}$, respectively. Moreover, the generated gaits are to be dynamically balanced.

The robot is checked for its dynamic balance using the concept of zero moment point (ZMP).

The robot is said to be dynamically balanced, when the ZMP lies inside the foot support polygon. The position of ZMP with respect to the ankle joint measured in the direction of motion as follows:

$$x_{ZMP} = \frac{\sum_{i=1}^{7}\left(I_i\,\dot{\omega}_i + m_i x_i\left(\ddot{z}_i - g\right) - m_i\,\ddot{x}_i\,z_i\right)}{\sum_{i=1}^{7} m_i\left(\ddot{z}_i - g\right)}, \quad (4)$$

where Ii denotes the moment of inertia of i^{th} link (kg-m2), $\dot{\omega}_i$ is the angular acceleration of link i in (rad/s2), mi denotes the mass of i^{th} link (kg), (xi, zi) is the coordinate of i^{th} lumped mass, g is the acceleration due to gravity (m/s2), $\ddot{z}_i$ is the acceleration of link i in z-direction (m/s2), $\ddot{x}_i$ is the acceleration of link i in x-direction (m/s2). If this ZMP is seen to lie outside the foot support polygon, the configuration of the robot is to be updated to move the ZMP to the support polygon. Dynamic balance margin (DBM) is calculated as the distance of ZMP from the boundary of support polygon as follows:

$$DBM = \left(\frac{L_7}{2} - \left|x_{ZMP}\right|\right), \quad (5)$$

where L7 is the length of the supporting foot and x_{ZMP} represents the distance of ZMP from the ankle joint in the direction of motion.

Torque required at each joint of the robot for its locomotion has been determined using Lagrange formulation. Fig. 3 shows the D-H parameters setting. The relationship between the generalized parameters (q_i) and joint angles (θ_i) are as follows:

$$q_1 = \theta_1;\ q_2 = (90 - \theta_2);\ q_3 = (\theta_2 - \theta_3);\ q_4 = (\theta_3 - \theta_4);$$
$$q_5 = (\theta_4 - \theta_5);\ q_6 = (\theta_5 - \theta_6);\ q_7 = (\theta_7 - (90 - \theta_6)).$$

The generalized angles are assumed to follow fifth-order polynomials in order to ensure their smooth variations, as given below:

$$q_i(t) = a_{i0} + a_{i1}t + a_{i2}t^2 + a_{i3}t^3 + a_{i4}t^4 + a_{i5}t^5 \quad (6)$$

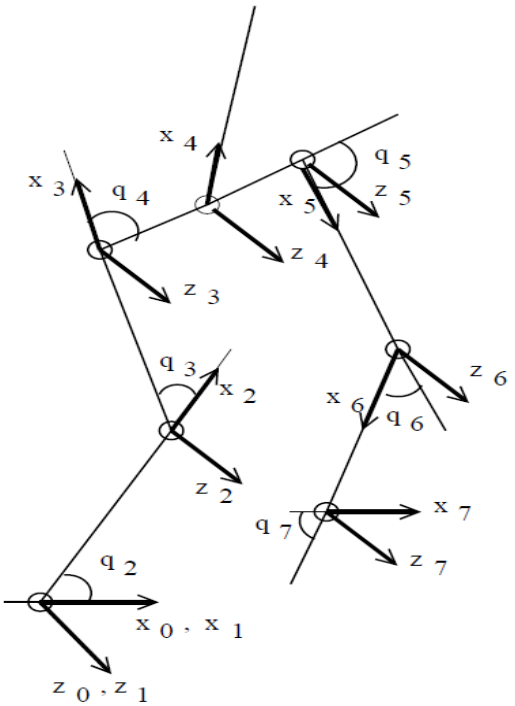


Fig. 3 A schematic view showing D-H parameters setting.

where i= 1, 2,. . . . . , n joints and a_{i0} a_{i1} a_{i2} a_{i3} a_{i4} a_{i5}, are the coefficients, whose values are to be determined using some known conditions. The angular velocity and acceleration can be determined by differentiating $q_i(t)$ (refer to equation (6)) with respect to time once and twice, respectively.

The equations of joint torques can be written as follows:

$$\tau_i = \sum_{k=1}^{n} D_{ik}\ddot{q}_k + \sum_{k=1}^{n}\sum_{m=1}^{n} h_{ikm}\dot{q}_k\dot{q}_m + C_i \quad , \iota=1, 2. . . , \nu, (7)$$

$$D_{ik} = \sum_{j=\max(i,k)} Tr(U_{jk} J_j U_{ji}^T), \quad \iota, \kappa = 1, 2,. . . , \nu, (8)$$

$$h_{ikm} = \sum_{j=\max(i,k,m)} Tr\left(U_{jkm} J_j U_{ji}^T\right) \quad ,\iota, \kappa, \mu = 1, 2. . ., \nu, (9)$$

$$C_i = \sum_{j=1}^{n}\left(-m_j g U_{ji} \bar{r}_j^j\right), \quad \iota =1, 2. . ., \nu \qquad (10)$$

where Dik denotes inertia terms, hikm represents the Coriolis and centrifugal terms, and Ci indicates information of the gravity terms.

The amount of power consumed by i^{th} joint can be calculated as the product of motor torque and angular velocity. If the amount of heat loss of the motor is considered, the average power consumption over a cycle of time period T, is calculated as follows:

$$P_i = \frac{1}{T}\sum_{i=1}^{n}\int_0^T \left(\left|\tau_i \dot{q}_i\right| + K\tau_i^2\right) dt, \qquad (11)$$

where K is a constant, whose value has been assumed to be equal to 0.025 [26].

B. Staircase Descending

Fig. 4 displays the schematic view of a biped robot descending the staircase. The trajectory followed by the swing leg has also been assumed to be a cubic polynomial, which satisfies the following boundary conditions:

$$\alpha\tau \quad x = 0, z = 0;$$

$$\alpha\tau \quad x = -x_1 - f_s/2, z = 0;$$

$$\alpha\tau \quad x = -s_w - x_1 - f_s/2, z = -s_h + f_h/2;$$

$$\alpha\tau \quad x = -2s_w - x_1 + x_3, z = -2s_h.$$

The hip trajectory has been assumed in the similar way, as it has been done for the ascending case. It is important to note that a similar expression given in equation (4) can be used for determination of ZMP, but g has to be replaced by –g in this expression.

The generalized angles are to be determined as given below.

$$q_1 = \theta_1;\ q_2 = (-90 + \theta_2);\ q_3 = (\theta_3 - \theta_2);\ q_4 = (\theta_4 - \theta_3);$$

$$q_5 = (\theta_5 - \theta_4);\ q_6 = (\theta_6 - \theta_5);\ q_7 = (\theta_7 - (90 + \theta_6)).$$

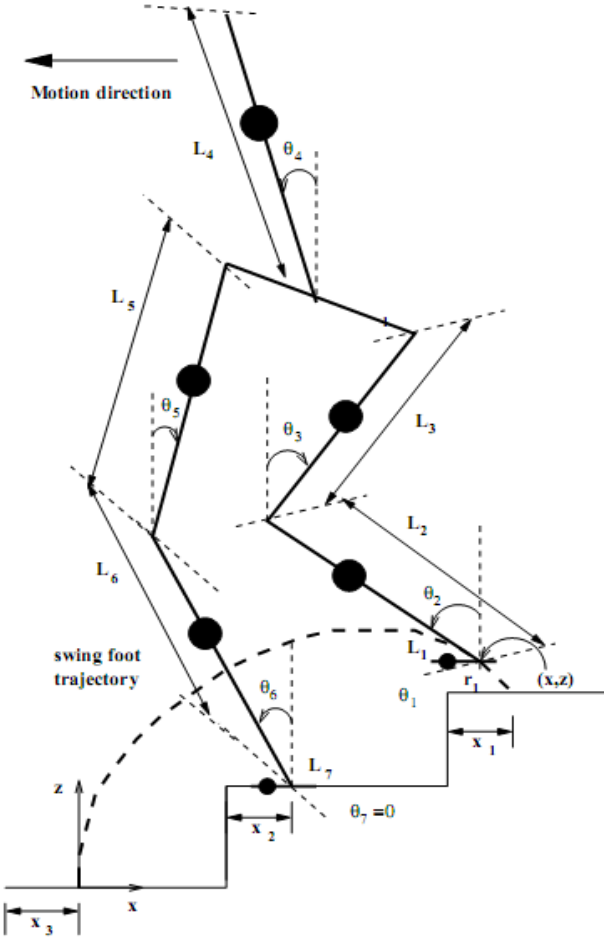


Fig. 4 A schematic view showing descending of a 7-dof biped robot.

The aim of this study is to minimize power consumption after keeping a maximum value of dynamic balance margin. As these two objectives contradict one another, Pareto-optimal front of solutions may exist. Thus, this problem may be posed as a multi-objective optimization problem for a biped robot generating gaits for ascending and descending the staircase. In the present study, both unconstrained and constrained optimization problems have been solved as stated below.

Case A: Unconstrained gait planning for ascending and descending staircase

This problem may be mathematically stated as follows:

Minimize average power consumption

$$P_i = \frac{1}{T}\sum_{i=1}^{n}\int_0^T \left(\left|\tau_i \dot{q}_i\right| + K\tau_i^2\right) dt,$$

and Minimize 1/DBM

$$1/DBM = 1/\left(\frac{L_7}{2} - |x_{ZMP}|\right),$$

subject to

$$r_1^{\min} < r_1 < r_1^{\max}; r_2^{\min} < r_2 < r_2^{\max}; r_3^{\min} < r_3 < r_3^{\max};$$

$$r_4^{\min} < r_4 < r_4^{\max}; m_4^{\min} < m_4 < m_4^{\max}.$$

The parameters: r_1, r_2, r_3 and r_4 denote the mass center positions of first, second, third and fourth links, respectively and m_4 represents the trunk mass. T indicates cycle time and n represents the number of joints. Due to symmetry of the biped robot, r_5, r_6 and r_7 have been kept equal to r_1, r_2 and r_3, respectively.

Case B: Constrained gait planning for ascending and descending staircase

In order to avoid jerky motion of the robot, the change in joint torque at each step of a cycle should be less than some pre-specified small value. It has been considered as functional constraint of this optimization problem.

This problem may be stated as follows:

Minimize average power consumption

$$P_i = \frac{1}{T}\sum_{i=1}^{n}\int_0^T \left(|\tau_i \dot{q}_i| + K\tau_i^2\right) dt,$$

and Minimize 1/DBM

$$1/DBM = 1/\left(\frac{L_7}{2} - |x_{ZMP}|\right),$$

subject to

$$\Delta\tau_{ij} \le \Delta\tau_{specified},$$

and

$$r_1^{\min} < r_1 < r_1^{\max}; r_2^{\min} < r_2 < r_2^{\max}; r_3^{\min} < r_3 < r_3^{\max};$$

$$r_4^{\min} < r_4 < r_4^{\max}; m_4^{\min} < m_4 < m_4^{\max}.$$

Here, $\Delta\tau_{ij}$ represents the change in torque of i^{th} joint at j^{th} time interval. It is to be noted that a violation of this constraint changes the torque requirement of the joint suddenly. Due to this sudden change in torque requirement, the motor connected to this joint may be overloaded and consequently, it may fail.

III. PROPOSED ALGORITHMS

Gait planning problems of the biped robot ascending and descending the staircase have been solved using two ANFIS models [27, 28]. The positions of two feet on the staircase, that is, x_1 and x_2 (refer to Fig. 1) are taken as inputs to the first ANFIS model (refer to Fig. 5). Fig. 6 displays an ANFIS model involving two inputs and outputs each. The membership function distributions of two variables: x_1 and x_2 are shown in Fig. 7. As four linguistic terms are used to represent each of these variables, there are 4x4=16 combinations of input variables, and thus, 16 rules. According to Takagi and Sugeno's approach, the j^{th} output of i^{th} rule is determined as $y_{ij} = a_{ij}x_1 + b_{ij}x_2 + c_{ij}$; where $i = 1,2,...,16; \quad j = 1,2;$ a_{ij}, b_{ij} and c_{ij} are the coefficients varied in the range of (0.0,1.0). The hip height h_1 and l_1 are determined as the outputs of this module of ANFIS. During optimization, the half base-widths of triangular membership functions (that is, $a_1^{'}$ and $a_2^{'}$ of Fig. 7) are optimized in the range of (0.001 to 0.025 m). Using the information of h_1 and l_1, the changes in angles: θ_2 and θ_3 can be determined analytically as discussed above. These changes (that is, $\delta\theta_2$ and $\delta\theta_3$) are used as inputs to the second module of ANFIS, which produces two outputs, namely the changes in θ_1 and θ_4, that is, $\delta\theta_1$ and $\delta\theta_4$. Fig. 8 shows the membership function distributions of two inputs of second module of ANFIS. As four linguistic terms are used to represent each input variable, there is a set of 16 rules for this module of ANFIS also. The j^{th} output of i^{th} rule can be written as follows: $Y_{ij} = d_{ij}\delta\theta_2 + e_{ij}\delta\theta_3 + f_{ij}$, where $i = 1,2,...,16; \quad j = 1,2;$ d_{ij} e_{ij} and f_{ij} are the coefficients to be determined during optimization, and these coefficients are varied in the range of (0.0,1.0). The values of two parameters: $a_3^{'}$ and $a_4^{'}$ have been varied in the range of (0.01, 7.0) and (0.01, 15.0), respectively. It is to be mentioned that the values of the above coefficients are varied in the ranges of (0.0, 1.0) during the training of the ANFIS models. It is also important to note that during the optimization of first module of ANFIS only, the number of design variables becomes equal to 2x48 (that is, coefficients of rules for two different outputs) + 2 (representing half base-widths of triangular memberships of two inputs) +5 (that is, r_1, r_2, r_3 r_4 m_4) =103. Similarly, for the second module of ANFIS, the optimization tool will have to tackle 2x48+2=98 more variables. Thus, a total of 103+98 = 201 variables are to be dealt with for tuning of two ANFIS modules. This optimization problem has been solved using two techniques separately, namely Genetic Algorithm

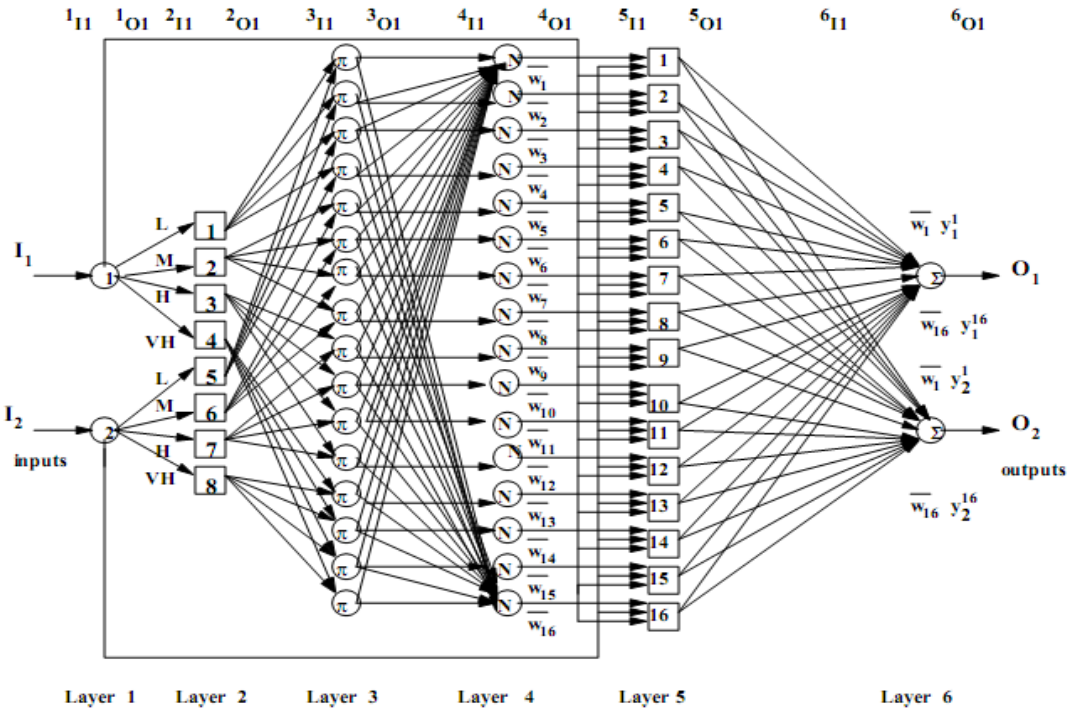


Fig. 6 A schematic view of an ANFIS for two inputs and two outputs used in gait planning.

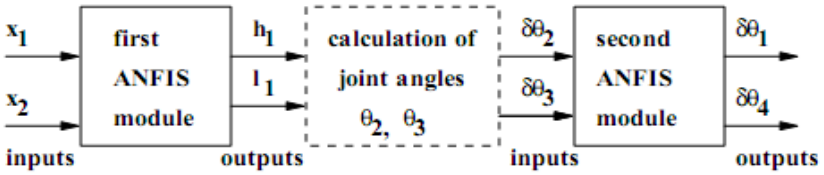


Fig.5 A schematic view of two ANFIS modules considered for biped gait planning.

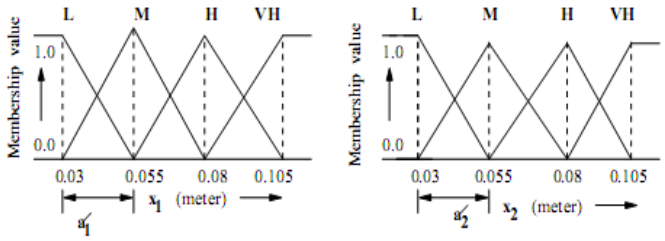


Fig. 7 Membership functions of inputs to first ANFIS module.

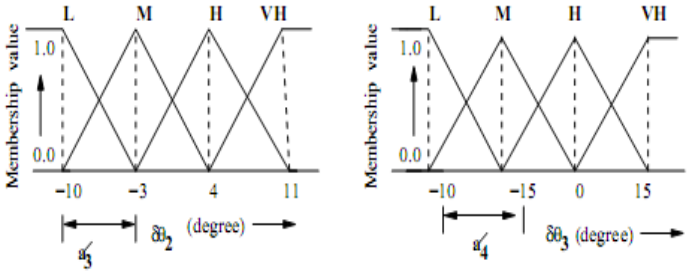


Fig. 8 Membership functions of inputs to second ANFIS module.

(GA) [20] and Particle Swarm Optimization (PSO) [32], separately.

Genetic Algorithm

Genetic Algorithm (GA) is a population-based probabilistic search technique, which works based on the principle of natural genetics. The genetics itself relies on the survival of the fittest. The evolution occurs through reproduction, crossover, and mutation. The GA has been used for solving multi-objective optimization problem also. Non-dominated Sorting Genetic Algorithm (NSGA-II) is one of such examples.

Initially, a random population of solutions of particular size is generated. Each solution is compared with others lying in the population in terms of fitness to find, if it is dominated. In this regard, two entities are calculated, namely (i) domination count np, i.e., the number of solutions which dominate the solution p, and (ii) sp, a set of solutions that the solution p dominates. Now, using the concept of np and sp, the whole population of solutions is divided into some fronts like first, second, third, and so on.

In order to preserve the good spread in the obtained set of solutions, two concepts, namely crowding distance and

crowding comparison are utilized. A crowding comparison is used to test as density-estimation metric, that is, to get an estimation of the density of solutions surrounding a particular solution in the population. The average distance of two points on either side of this point along each of the objectives is calculated. This distance is nothing but the perimeter of the cuboid formed using the nearest neighbors as the vertices, which is known as crowding distance. The crowding distance values are used for sorting the population according to each objective function value in ascending order of magnitude. Therefore, for each objective function, the boundary solutions (solutions with the smallest and largest function values) are assigned an infinite distance value. All other intermediate solutions are assigned distance values equal to the absolute normalized differences in the function values of two adjacent solutions. This calculation is continued with other objective functions. The overall crowding distance value is calculated as the sum of the individual distance value corresponding to each objective. Each objective function is normalized before determining the crowding distance. For all solutions in the non-dominated set, initialize distance for each solution to zero. For each objective function (fitness) value, sort all other non-dominated values in descending order. The crowded-comparison selection process at the various stages of the algorithm guides towards uniformly spread-out Pareto-optimal front using non-domination rank and crowding distance. For two solutions with differing non-domination ranks, the one with the lower rank will be selected. Otherwise, if both solutions belong to the same front (that is, rank) the solution located in a less crowded region is selected.

Finally, the complete algorithm can be summarized as follows:

1) Create the population of solutions at random.

2) Sort all solutions into some non-dominated fronts (also known as ranks). Calculate the crowding distance as mentioned above for each solution of the non-dominated front.

3) Use GA operators for creating the population of solutions for the next generation.

4) Increase the generation counter.

This process is repeated until the number of generation reaches the pre-specified maximum number of generations. NSGA-II algorithm [20] is also capable of handling constrained optimization problems. A fixed penalty value has been used, if there is a violation of any constraint. Interested readers may refer to [20] for a detailed description of the algorithm.

In the present problem, a total of 201 variables have been coded in the real-coded GA. The GA will try to determine the solution corresponding to the maximum dynamic balance margin of the robot but at the expense of minimum power. The power consumption and dynamic balance margin of the robot have been calculated using equations (11) and (5), respectively. This problem has been formulated as a multi-objective optimization one (as discussed in Section 2) and accordingly, the fitness values of GA-solutions have been calculated.

Particle Swarm Optimization Algorithm

Particle Swarm Optimization (PSO) is a technique, which works based on the concept of swarm behavior in searching food in the neighborhood space efficiently and effectively. The particle is defined with respect to its two parameters, namely position and velocity in search space. The i^{th} particle's position and velocity vectors in d -th dimensional search space can be represented as $X_i = (x_{i1}, ..., x_{id})$ and $V_i = (v_{i1}, ..., v_{id})$, respectively. The value of V_i vector can be varied in the range of $[-v_{\max}, v_{\max}]$ to reduce the tendency of particles to leave the search space. The value of $v_{\max}$ is usually chosen to be equal to $k \times x_{\max}$, where $0.1 \le k \le 1.0$ [30]. The trajectory of each individual in the search space is adjusted dynamically according to its own flying experience and information provided by other particles in the search space. The swarm's global best solution is achieved simply by adjusting the trajectory of each individual toward its own best location and the best particle of the entire swarm at each time step (generation) [17, 29]. For a given fitness function, each particle's best solution (Pbest) at time t given as $P_i = (p_{i1}, ..., p_{id})$, and the swarm's fittest particle (Gbest) during the same time t is denoted by $P_g = (p_{g1}, ..., p_{gd})$. The new velocities and positions of the particles for the next fitness evaluation are calculated using the following two equations:

$$v_{id}(t+1) = W\, v_{id}(t) + b_1 rand(\cdot)(p_{id} - x_{id}(t)) + b_2 Rand(\cdot)(p_{gd} - x_{id}(t)), \quad (12)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1), \quad (13)$$

where v_{id} is the velocity of d -th dimension of i^{th} particle, W is a constant known as inertia weight [31], b_1 and b_2 denote the acceleration coefficients, and $rand(\cdot)$ and $Rand(\cdot)$ are two separately generated uniformly distributed random numbers lying in the range of $[0,1]$. The first part of equation (12) denotes the previous velocity that provides the necessary momentum to the particles to move across the search space. The second part of the equation (12) represents cognitive component responsible for personal thinking of each particle. It helps the particles to move toward their respective best solutions. The third part of equation (12) indicates social component, which controls collaborative effect of the particles in order to find the globally best solution. The PSO is able to gain much attention nowadays due to its simple architecture,

ease of implementation and ability to quickly reach the global optimal solution.

Multi-objective PSO (that is, MOPSO-CD) [32, 33] incorporates the mechanism of crowding distance of NSGA-II into the PSO. The concept of crowding distance together with mutation operator maintains the diversity of non-dominated solutions. The MOPSO-CD can also handle constrained optimization problems. The working principle of MOPSO-CD can be stated in steps as follows:

1. Randomly generate a population of particles, whose velocities are set equal to zero.

2. Compute each particle's best fitness: Pbest, swarm's best fitness, namely Gbest. Store Pbest and Gbest solutions in archive. Compute the crowding distance in the archive and sort these values in descending order. Randomly select a solution as Gbest one from the top (10%) of the values. Gbest acts as the global best guide for non-dominated solutions.

3. Update velocities and positions of the population of particles.

4. Insert non-dominated solutions obtained from the updates into the archive. Delete all dominated solutions from the archive. Compute the crowding distance and sort in descending order of the values. Select randomly the most crowded values (bottom 10%) as the particle's Pbest. Update the particle's Pbest.

5. Repeat iterations.

As discussed above, the values of two objectives, namely power consumption and dynamic balance margin are dependent on 201 variables, which have been coded in the PSO-solutions. The fitness values of PSO-solutions have been determined corresponding to the said two objectives, as these are calculated for the GA-solutions. A penalty function approach has been adopted in order to penalize a solution, if there is a violation of functional constraint.

IV. RESULTS AND DISCUSSION

The biped robot consists of links having the masses (in kg): $m_1 = m_7 = 0.5$; $m_2 = m_6 = 2.0$; $m_3 = m_5 = 5.0$ and m_4 is varied in the range of 10.0 to 50.0 kg. The links are assumed to have the lengths (in m) as follows: $l_1 = l_7 = 0.06$, $l_2 = l_6 = 0.34$, $l_3 = l_5 = 0.30$, $l_4 = 0.6$. The cycle time t has been assumed to be equal to 5.0 seconds. The maximum velocity of the swing foot has been considered to be equal to 0.056 m/s. In simulations, the values of $r_1^{\min}$ $r_1^{\max}$ $r_2^{\min}$ $r_2^{\max}$ $r_3^{\min}$ $r_3^{\max}$ $r_4^{\min}$ $r_4^{\max}$ $m_4^{\min}$ and $m_4^{\max}$ have been set as 0.01, 0.02, 0.1, 0.32, 0.1, 0.28, 0.1, 0.54, 10.0 and 50.0, respectively. Computer simulations are carried out on a P–IV PC. Results related to unconstrained and constrained optimizations are stated and discussed below.

A. Results of Unconstrained Optimization

Results of unconstrained multi-objective optimization problems related to ascending and descending the staircases, as obtained by the GA and PSO algorithm separately, have been stated and discussed below.

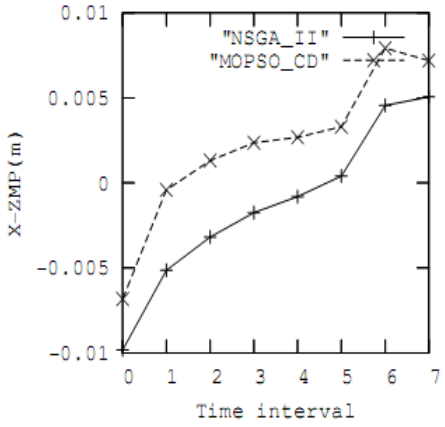
1) Ascending the staircase

As the performance of both the GA and PSO algorithm are dependent on their parameters' values, a separate thorough parametric study has been carried out for each of these algorithms. In this study, only one parameter has been varied at a time, keeping the others fixed. The following GA-parameters are found to yield the best results: crossover probability p_c = 0.8; mutation probability p_m = 0.00505; maximum number of generations = 100 and population size = 100. Similarly, the following PSO parameters are seen to give the best results: number of runs = 100; swarm size = 100. Fig. 9(a) displays the variations of x_{ZMP} in a cycle as obtained by the GA- and PSO-based optimization. It indicates that the robot is able to maintain its dynamic balance in the cycle using the optimized parameters yielded by the GA and PSO algorithm, separately. The variations of joint angles in a cycle corresponding to the optimal parameters determined by the GA and PSO algorithm are shown in Fig.9 (b). Moreover, the positions of various link masses in X-Z plot at different instants in a cycle have been determined for the set of optimized parameters decided by the GA and PSO algorithm, separately (refer to Fig 9(c)). Fig. 10 displays Pareto-optimal fronts of solutions obtained by the GA and PSO algorithm for the staircase ascending problem. It is interesting to observe that PSO algorithm has yielded a better Pareto-optimal front of solutions in comparison with that obtained by the GA. It has happened so, due to the reason that the PSO algorithm can carry out both the global and local searches simultaneously; whereas the GA is weak in local search, although it is a potential tool for global optimization. Moreover, the particles (solutions) in PSO algorithm rely on their memories also, while moving from one population to the next, which is missing in the GA-search. It is to be noted that one generation of the PSO and GA has taken 769.94 and 812.33 seconds, respectively, as computational time during the optimization. Any point lying on this front is an optimal solution obtained after considering a particular set of weights on two objective functions. Thus, the designer may have a choice to select the suitable optimal solution out of all the points lying on the Pareto-optimal front.

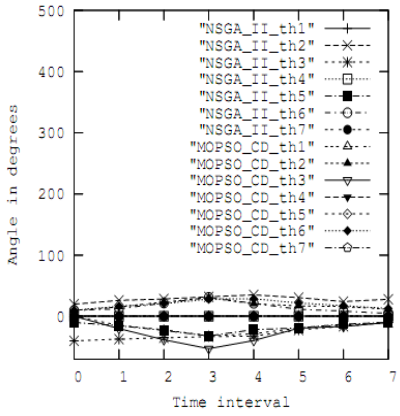
2) Descending the staircase

The optimized GA- and PSO-parameters have been determined separately as discussed above for the problems of staircase descending also. The best results are obtained with the following GA-parameters: crossover probability p_c = 0.8, mutation probability p_m = 0.00505, maximum number of generations = 100 and population size = 100. Moreover, the following PSO-parameters have yielded the best results: number of runs =100, swarm size =100. Fig 11 displays the variations of x_{ZMP}, joint angles and various positions of the link masses at different instants in a cycle. The Pareto-optimal

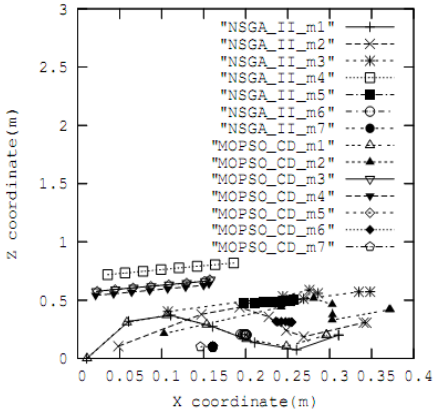
fronts of solutions obtained by the GA and PSO algorithm separately, for this problem, are shown in Fig. 12. It is important to mention that the PSO has taken 805.54 seconds time to complete its one generation, whereas the GA has taken a slightly more time (890.31 seconds) for the same. Once again, the GA has been defeated by the PSO algorithm in terms of the quality of Pareto-optimal front of solutions. It may be due to the reasons mentioned above. It is important to mention that the minimum power consumption of the robot, as indicated in Fig. 12, is found to be equal to 0.03W.



(a)



(b)



(c)

Fig. 9 Variations of (a) x_{ZMP}, (b) joint angles, and (c) positions of the masses in a cycle, as obtained by the GA and PSO algorithm for unconstrained optimization in staircase ascending problems.

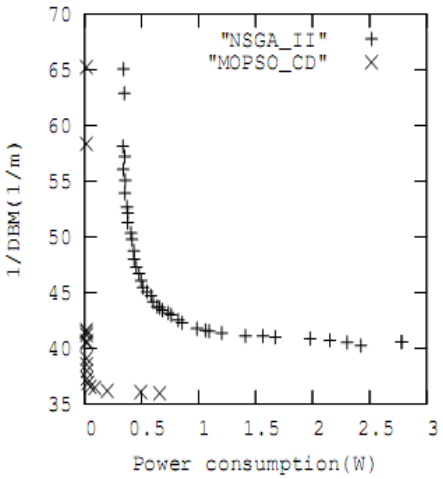
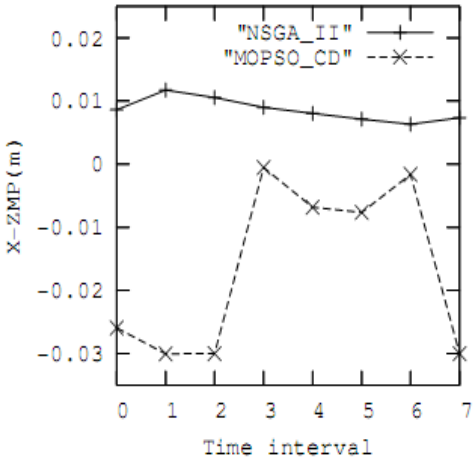
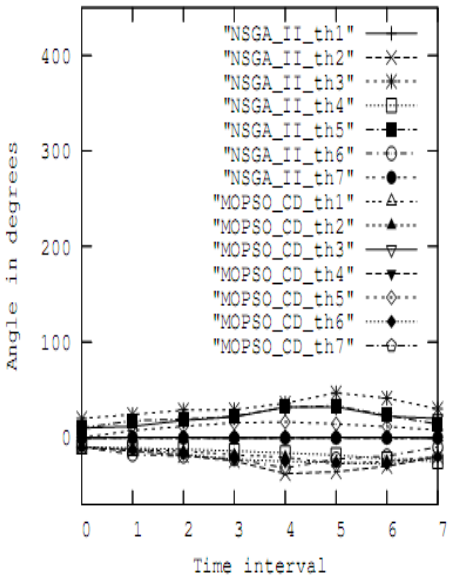


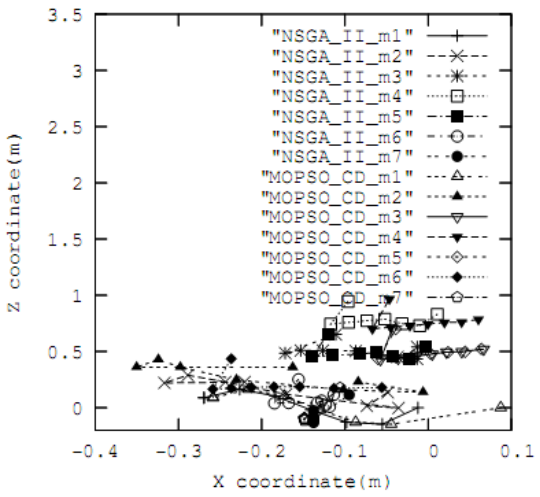
Fig. 10 Pareto-optimal fronts of solutions obtained by the GA and PSO algorithm for unconstrained optimization in staircase ascending problems.



(a)



(b)



(c)

Fig. 11 Variations of (a) x_{ZMP}, (b) joint angles, and (c) positions of the masses in a cycle, as obtained by the GA and PSO algorithm for unconstrained optimization in staircase descending problems.

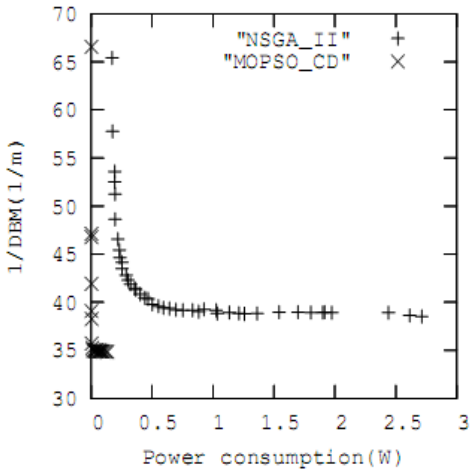


Fig. 12 Pareto-optimal fronts of solutions obtained by the GA and PSO for unconstrained optimization in staircase descending problems.

3) Discussion

The left bottom-most point of Figs.10 and 12 corresponds to minimum power consumption and high DBM, whereas the maximum values of power consumption and DBM are indicated by the right-most point of the said figures. Moreover, the trunk mass is found to reach its minimum and maximum values at the left- and right-most points of the Pareto-optimal front of solutions, respectively. Thus, both power consumption as well as DBM is seen to increase with the trunk mass. It is in line with the observations of human-beings ascending and descending the staircases.

A close watch on Figs.10 and 12 reveals that for a particular value of power consumption, DBM is more in descending gait compared to that in ascending gait. Moreover, ascending gait is found to consume more power in comparison with the descending gait for a particular value of DBM. These observations also match with the general experiences of human-beings.

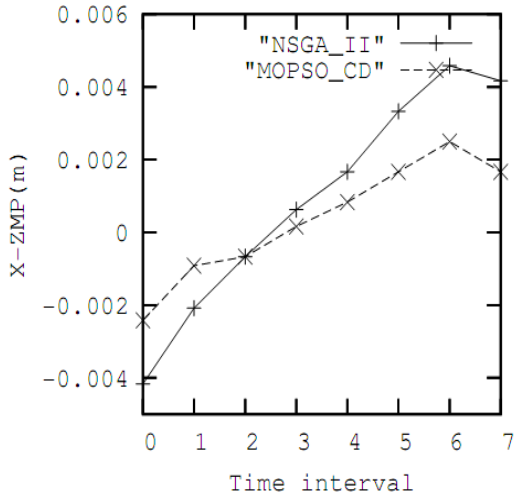
The performances of the GA and PSO algorithm can also be compared through Figs. 10 and 12 in terms of the quality of Pareto-optimal fronts obtained by these two algorithms. It is interesting to observe that the PSO has obtained the better Pareto-optimal fronts of solution compared to that achieved by the GA in both the ascending and descending cases. Moreover, the PSO is found to be faster than the GA. It has happened so, due to the reasons discussed above. Moreover, the PSO is a greedier algorithm compared to the GA.

B. Results of Constrained Optimization

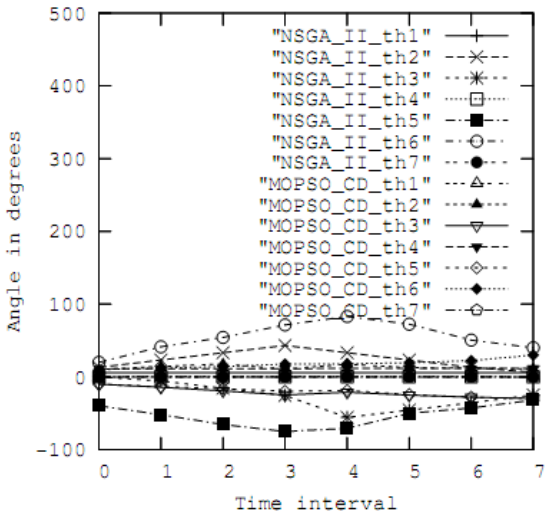
Constrained optimization problems related to ascending and descending the staircases have also been tackled using the GA and PSO algorithm, separately. It is important to mention that a penalty function approach has been adopted to penalize a solution, if there is a violation of constraint. The results are stated, discussed and compared below.

1) Ascending the staircase

The following GA-parameters have given the best results: crossover probability p_c = 0.8, mutation probability p_m = 0.00505, maximum number of generations = 100 and population size = 100. Similarly, the following PSO-parameters obtained through a careful study are seen to yield the best results: number of runs=100, swarm size =100. The variations of x_{ZMP}, joint angles and positions of the masses in a cycle are displayed in Fig. 13. Fig. 14 shows the Pareto-optimal fronts of solutions as obtained by the GA and PSO algorithm, for this problem. In order to complete one generation, the PSO and GA are found to take 785.15 and 862.14 seconds, respectively. The PSO algorithm has outperformed the GA again, and the reasons behind this fact have been explained above. Any point lying on this front is an optimal solution obtained after assigning a particular set of weights on two objective functions. Thus, the designer will be able to select a suitable optimal solution out of all the points lying on the Pareto-optimal front.



(a)



(b)

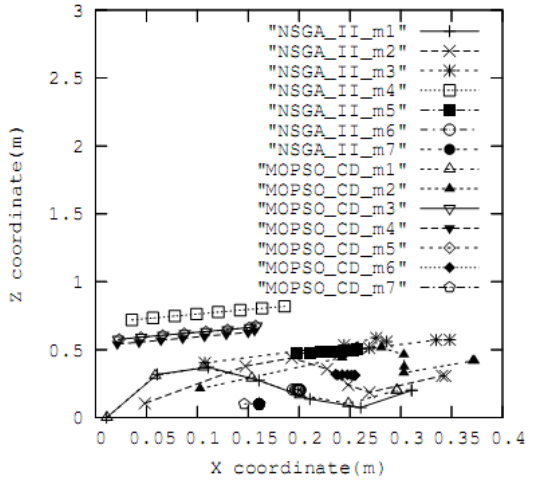


Fig. 13 Variations of (a) x_{ZMP}, (b) joint angles, and (c) position of the masses in a cycle, as obtained by the GA and PSO algorithm for constrained optimization in staircase ascending problems.

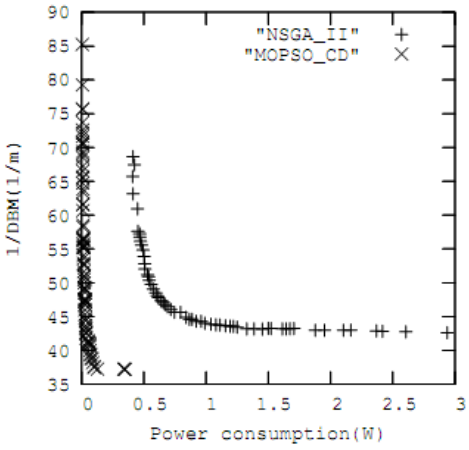
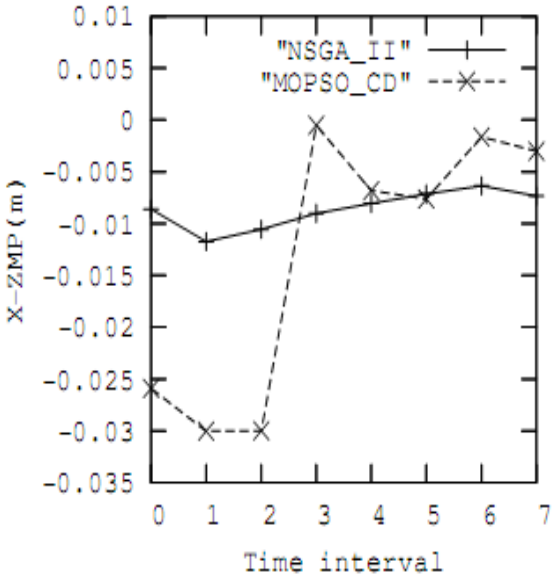


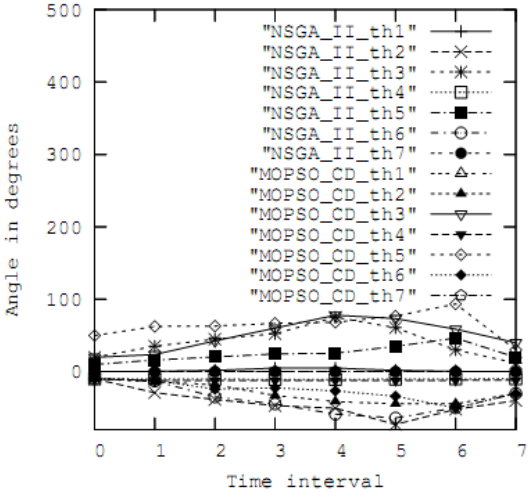
Fig. 14 Pareto-optimal fronts of solutions obtained by the GA and PSO algorithm for constrained optimization in staircase ascending problems.

2) Descending the staircase

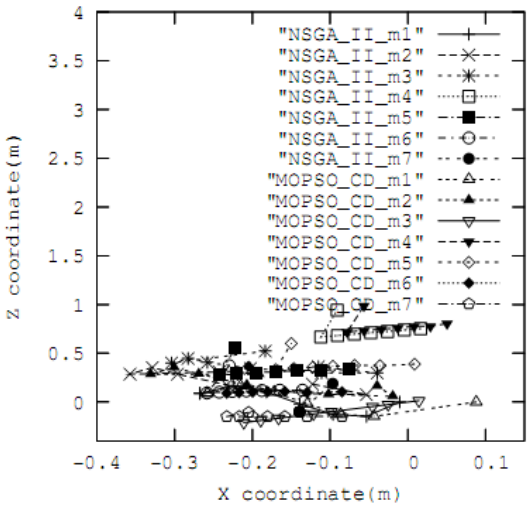
The optimal set of GA-parameters has been obtained as follows: crossover probability p_c = 0.8, maximum number of generations = 100 and population size = 100. The optimized PSO-parameters are found to be like the following: number of runs=100, swarm size =100. Fig. 15 displays the variations of x_{ZMP}, joint angles and positions of the masses with time in a cycle. These variations are found to be significantly different from those shown in Fig. 13. The Pareto-optimal fronts of solutions achieved by the GA and PSO algorithm are shown in Fig. 16. The PSO has taken 818.25 seconds to complete its one generation, whereas the GA is seen to take 890.56 seconds for the same. It is to be noted that the PSO algorithm has achieved optimal solution corresponding to a minimum power consumption of 0.0131W. Once again, the GA has been defeated by the PSO algorithm in terms of the quality of obtained Pareto-optimal fronts of solutions.



(a)



(b)



(c)

Fig. 15 Variations of (a) x_{ZMP} , (b) joint angles, and (c) positions of the masses in a cycle, as obtained by the GA and PSO algorithm for constrained optimization in staircase descending problems.

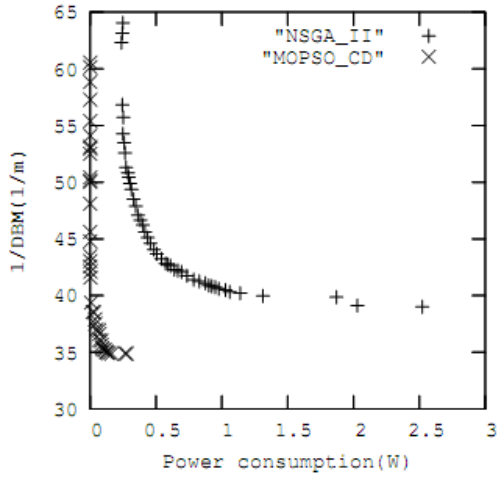


Fig. 16 Pareto-optimal fronts of solutions obtained by the GA and PSO algorithm for constrained optimization in staircase descending problems.

3) Discussion

A comparison between Figs. 14 and 16 indicates that the DBM in descending gait becomes more than that in ascending gait generation for a particular value of power consumption. Moreover, ascending gait requires more power than descending gait does for a particular value of DBM. As expected, power requirement in staircase ascending has turned out to be more than that in descending (refer to Figs. 10 and 12, and Figs. 14 and 16). Once again, the PSO is found to outperform the GA in terms of the quality of obtained Pareto-optimal fronts of solution. Moreover, the PSO is seen to be faster than the GA during the optimization. It has happened so, for the reasons mentioned above.

V. CONCLUDING REMARKS

This study deals with gait planning problems of a 7-dof biped robot ascending and descending some staircases after consuming minimum power and maintaining a high dynamic stability margin. As these two objectives contradict each another, it is an ideal problem for multi-objective optimization. Both the constrained as well as unconstrained optimization problems have been solved using the GA and PSO algorithm separately, and Pareto-optimal fronts of solutions have been obtained. The findings of the study in terms of power consumption and dynamic stability margin are in tune with the general experience of human beings ascending and descending the staircases. The PSO algorithm has shown better performance compared to the GA. Moreover, the PSO algorithm is seen to be faster compared to the GA. The former carries out both the global and local searches simultaneously, whereas the latter is a potential tool for the global search only. The obtained Pareto-optimal fronts of solutions may help the designer to select some appropriate optimal solutions depending on the requirements. Thus, it helps the designer to arrive at a suitable design of the biped robot. In the present study, a simplified model of biped robot having 7dof has been considered, whose movement is restricted in the direction of its travel, that is, X direction. However, its movement in Y direction has been neglected for simplicity. A real humanoid robot has its movements in both X and Y directions, besides its motion along Z direction. Therefore, the present model is a simplified version of a real humanoid robot. However, the present analysis may be considered as an aid to carry out the similar analysis for a realistic humanoid robot. It has been kept in the scope of future work. For simplicity, the hip is assumed to follow a straight path having the slope equal to that of the staircase in order to maintain repeatability of the cycle. However, some other trajectories may also be tried for the hip joint and their impacts on power consumption and DBM will be studied in future. In the present study, only one functional constraint, that is, change in joint torques to be within a pre-specified range, has been considered. However, there may be some other functional constraints like acceleration limit and power rating of the motor, and others, which will be studied in future.

ACKNOWLEDGEMENTS

The authors wish to thank AICTE, India, for supporting this study.

REFERENCES

[1] M. Vukobratovic, A.A. Frank and D. Juricic, "On the stability of biped locomotion," IEEE Trans. on Biomedical Engineering, 17(1), pp. 25-36, 1970.

[2] S.G.Capi, K. Kaneko, Mitobe, Barolli , Y. Nasu., "Optimal trajectory generation for a prismatic joint biped robot using genetic algorithm," Robotics and Autonomous Systems, 38, pp. 119-128, 2002.

[3] G. Capi, , Y.Nasu,, L. Barolli, and K. Mitobe., "Real-time gait generation for autonomous humanoid robots: A case study for walking," Robotics and Autonomous Systems, 42, pp. 107-116, 2003.

[4] K.S. Jeon, O. Kwon, and J.H. Park., "Optimal trajectory generation for a biped robot walking a staircase based on genetic algorithms," Proc. of IEEE Intl. Conf. on Intelligent Robots and Systems, Sendai, Japan, pp. 2837- 2842, 2004.

[5] A.W. Salatian, and Y.F. Zheng., "Gait synthesis for a biped robot climbing sloping surfaces using neural networks. Part I. Static learning." Proc. of IEEE Intl. Conf. on Robotics and Automation, Nice, France, pp. 2601-2606, 1992.

[6] A.W. Salatian., and Y.F. Zheng., "Gait synthesis for a biped robot climbing sloping surfaces using neural networks. Part II. Dynamic learning." Proc. of IEEE Intl. Conf. on Robotics and Automation, Nice, France, pp. 2607-2611, 1992.

[7] S.Fan, M.Sun and M.Shi., "Real-time gait generation for humanoid robot based on fuzzy neural networks," Proc. of the Third Intl. Conf. on Natural Computation Haikou, Hainan, China, 2, pp. 343-348, 2007.
[8] A.Kun and W.T.Miller III., "Adaptive fuzzy systems of a biped robot using neural networks," Proc. of IEEE Intl. Conf. on Robotics and Automation, Minneapolis, Minnesota, USA, pp. 240-245, 1996.
[9] W.T.Miller III., "Real-time neural network control of a biped walking robot," IEEE Trans. on Control Systems, 14, pp. 41-48, 1994.
[10] C. Zhou and Q. Meng, Q., "Dynamic balance of a biped robot using fuzzy reinforcement learning agents," Fuzzy Sets and Systems, 134, pp. 189-203, 2003.
[11] R.K Jha, B. Singh and D.K. Pratihar., "Online stable gait generation of a two-legged robot using a genetic-fuzzy system," Robotics and Autonomous System, 53, pp. 15-35, 2005.
[12] A.D. Udai., "Optimum hip trajectory generation of a biped robot during single support phase using genetic algorithm," Proc. of First Intl. Conf. on Emerging Trends in Engineering and Technology, Nagpur, India, pp. 739-744, 2008.
[13] P.R Vundavilli, S.K.Sahu and D.K.Pratihar., "Online dynamically balanced ascending and descending gait generations of a biped robot using soft computing," Intl Jl. of Humanoid Robotics, 4(4), pp. 777- 814, 2007.
[14] J.Y.Lee and J.J. Lee., "Optimal walking trajectory generation for a biped robot using multi-objective evolutionary algorithm," Proc. of IEEE Control Conference, Melbourne, Australia, 1, pp. 357- 364, 2004.
[15] G. Capi, M. Yokota and K. Mitobe., "A new humanoid robot gait generation based on multi-objective optimization," Proc. of IEEE/ASME Intl. Conf. on Advanced Intelligent Mechatronics. Monterey, California, USA, pp. 450- 454, 2005.
[16] D. Goswami, V. Prahlad, and P.D Kien., "Genetic algorithm-based optimal bipedal walking gait synthesis considering trade-off between stability margin and speed," Robotica, 27, pp. 355–365, 2009.
[17] J. Kennedy and R. Eberhart, R., "Particle swarm optimization," Proc. of IEEE Intl. Conf. on Neural Networks, Perth, Australia, pp. 1942–1948, 1995.
[18] M.M. Millonas., Swarms, phase transitions, and collective intelligence. C. G. Langton, (ed.), Artificial Life III. Addison Wesley, Reading, MA, (1994).
[19] C.A.Coello Coello, G.T.Pulido and M.S. Lechuga., "Handling multiple objectives with particle swarm optimization," IEEE Trans. on Evolutionary Computation, 8(3), pp. 256-279, 2004.
[20] K.Deb, A.Pratap, S.Agarwal and T.Meyarivan., "A fast and elitist multiobjective genetic algorithm: NSGA-II," IEEE Trans. on Evolutionary Computation, 6(2), pp. 182-197, 2002.
[21] M.Reyes-Sierra and C. A.Coello Coello., "Multi-objective particle swarm optimizers: a survey of the state-of-the-art," Intl. Jl. of Computational Intelligence Research. 2(3), pp. 287– 308, 2006.
[22] R.Poli., "Analysis of the publications on the applications of particle swarm optimization," Hindawi Publishing Corporation, Jl. of Artificial Evolution and Applications, pp: 1-10, 2008.
[23] K.Sivakumar, C.Balamurugan,S. Ramabalan and S.B.Venkata raman., "Optimal concurrent dimensional and geometrical tolerancing based on evolutionary algorithms," Proc. of IEEE World Congress on Nature & Biologically Inspired Computing (NaBIC 2009), pp. 300-305. 9-11, December, Coimbatore, India, 2009.
[24] N. Rokbani, E.Benbousaada, B.Ammar and M. Alimi Adel., "Biped robot control using particle swarm optimization," Proc. of Intl. Conf. on Systems Man and Cybernetics (SMC), IEEE, pp. 506-512.,10-13 Oct, 2010, Istanbul, Turkey.
[25] C.Niehaus , T.R¨ofer., T.Laue., "Gait optimization on a humanoid robot using particle swarm optimization," Proc. of the Second Workshop on Humanoid Soccer Robots, IEEE-RAS, Intl. Conf. on Humanoid Robots, 2007. Pittsburgh, PA, USA, 2007.
[26] J.Nishii, K.Ogawa and R.Suzuki., "The optimal gait pattern in hexapods based on energetic efficiency," Proc .of the 3rd Intl. Symp. on Artificial Life and Robotics, 29 oct - 01 Nov, 1998, Hong Kong, pp. 106–109.
[27] J. R. Jang., "ANFIS: Adaptive-network-based fuzzy inference system," IEEE Trans. on Systems, Man and Cybernetics, Part B, 23(3), pp. 665-685, 1993.
[28] D.K.Pratihar, Soft Computing, Narosa Publishing House, New Delhi, India. 2008.
[29] M.Clerc and J.Kennedy., "The particle swarm-explosion, stability and convergence in a multi-dimensional complex space," IEEE Trans. on Evolutionary Computation, 6, pp. 58-78, 2002.
[30] R.C. Eberhert, R.C., P. Simpson,P., R.Dobbins, R., Computational Intelligence PC tools: Dalian, Academic Press, San Diego, USA. Ch 6, pp. 212-226, 1996.
[31] Y.Shi and R.C. Eberhart., "Empirical study of particle swarm optimization," Proc. of IEEE Intl. Congr. on Evolutionary Computation, 3, Washington D.C, USA, pp. 101-106, 1999.
[32] C.R.Raquel and P.C.Naval Jr., "An effective use of crowding distance in multi-objective particle swarm optimization," Proc. of Genetic and Evolutionary Computing Conference, 2005(GECCO '05), pp. 257-264, June 25-29, 2005, Washington DC, USA.
[33] (2010) http:// www.particleswarm.info/
[34] J.J.Kim, J.W. Lee and J.J.Lee., "Central pattern generator parameter search for a biped walking robot using nonparametric estimation based particle swarm optimization," Intl. Jl. of Control, Automation and Systems , 7(3), pp. 447-457,2009.