Int. J. Mach. Learn. & Cyber.
DOI 10.1007/s13042-017-0660-1

ORIGINAL ARTICLE

Multiobjective optimized bipedal locomotion

Manish Raj[1] · Vijay Bhaskar Semwal[1] · G C Nandi[1]

Received: 11 September 2016 / Accepted: 8 March 2017

Abstract To develop the general analytical model for the optimization of trade-off Stability and Energy functions for biped humanoid robot is very tedious work. This paper presents a novel analytical method to develop the multi-objective function includes energy and stability functions. The challenge was to develop analytical model for stability and energy for the single support phase (SSP), double support phase (DSP) and the transition to SSP-DSP and vice versa. The energy function has been developed by unique approach of orbital energy concept and the stability function obtained by modifying the pre-existing Zero Moment Point (ZMP) trajectory (the trajectories which generated by the mathematical equations of ZMP). These functions are optimized using Real Coded Genetic Algorithm to produce an optimum set of walk parameters.The analytical results show that, when the energy function is optimized, the stability of the robot decreases. Similarly, if the stability function is optimized,the energy consumed by the robot increases. Thus, there is a clear trade-off between the stability and energy functions. Thus,we propose the Multi-Objective Evolutionary Algorithm to yield the optimum value of the walk parameters. The results are verified by NaO robot.This approach increases the energy efficiency of NaO robot by 67.05% and stability increases by 75%. Furthermore, this method can be utilized on all ZMP classed bipeds(HOAP, Honda robots).

Keywords Zero moment point · Center of mass · Multi-objective optimization · Bipedal locomotion · Pareto-optimal front

✉ Manish Raj
rajmanish.03@gmail.com

[1] Indian Institute of Information Technology, Allahabad, India

1 Introduction

Bipedal locomotion is still an open ended problem and poses great challenges to the researchers from various backgrounds. One of the reasons responsible for the sluggish development of bipedal machines, is the need for simultaneously achieving two objectives i.e. higher energy efficiency and greater stability, both of which are important attributes of an autonomous robot. The flat footed bipedal robots ensures greater stability, but at an expense of consuming more energy [1]. On the other hand, the point footed bipedal robots consumes significantly less amount of energy as compared to flat footed robots, sacrificing the stability concerns required for performing tasks, other than walking. Most of the humanoid robots today, are flat footed, because of the versatility of tasks being performed by them other than walking. The flat footed robots use the concept of Zero Moment Point (ZMP) [2] to ascertain stability during different phases of walking. From the above discussions, it can be inferred that there is a clear trade-off between the energy consumed by the robot and its stability issues. That is, the higher the energy consumed by the robot, more stable it is and vice versa. Therefore, current research trends towards building efficient algorithms that balances the trade-off between the two contrasting objectives.

In the past, there has been significant contributions in developing energy efficient bipedal robots and optimizing their gait parameters using evolutionary algorithms. It was seen that, the demerit of the robots employing ZMP is that, they consume a lot of energy. The basic reason behind the enormous consumption of energy is, the large gap between the artificial gait synthesized in bipeds and the natural

Published online: 31 March 2017

human gait. The amount of energy consumed during walking depends on the type of gait pattern of a person [3].[1] The gait pattern plays a vital role in bipedal walking. This is because, it determines the optimal footholds, velocity and acceleration of the support foot. This, in turn affects the robot's dynamic stability, energy consumption, harmony and so on. In short, the quality of the legged locomotion is decided by its gait pattern [4]. Thus, carefully designing the gait pattern of the bipeds may help a lot, in reducing the energy consumption as well as increasing its stability. This research carried out in the field of gait optimization, in order to decrease the energy consumed by the robot as well as increase its stability.

Despite the huge time taken by the evolutionary algorithms for optimization, the gait optimization is generally performed by using genetic algorithm, due to its robustness in search and optimization problems [5]. This is because, most of the ZMP classed bipedal robots have offline walking pattern generation. So, in order to design an energy efficient gait, the optimization is carried out before the actual walking starts. Some of the contributions related to gait optimization is worth mentioning here. Capi et al. [6] used Real Coded Genetic Algorithm (RCGA) to optimize the energy consumed by the robot. The objective was to find the joint trajectories, which consumes minimum energy. The energy function was formulated by integrating the torque generated at the motor joints. It was a single objective optimization problem, where only energy was optimized. The stability criterion was used for imposing constraint on the energy function. That is, if the position of ZMP goes outside the support polygon, the energy function was heavily penalized. The drawback of this research was that, it did not explicitly specify the ZMP trajectory, which accounts for the biped's stability. Nasu et al. [7] synthesized the gait pattern of a biped, using Genetic Algorithm (GA) and Radial Basis Function Neural Network (RBFNN). The approach used GA first, to find the optimal set of gait parameters, which was then used to train the RBFNN, in order to generate real time energy efficient gait pattern. However, due to the large computational overhead, this method suffered significant delays in real time applications. Jong et al. [8] used GA to minimize energy consumed by a 6- degree of freedom (DoF) bipedal robot, by selecting optimal locations of the mass centers of the links. However, this method requires one to design a bipedal robots hardware, in such a manner that, the locations of center of mass (CoM) of each link corresponds to optimal locations of CoM previously found, using GA. Choi et al. [9] used GA to optimize the walking trajectory of IWR-III bipedal robot by minimizing the sum of deviations of velocities, accelerations and jerks in order to maintain continuity of joint trajectories and energy distribution at the via points. But, this research does not show, how implementing continuity in the joint trajectory results in the improvement of energy consumption. An analysis of [10] reveals that, variation of the walk parameters as well as the stiffness values of the joints affects the energy consumed by NaO. Experiments were performed on reducing the knee flexion and stiffness values of the joints. However, a formal mathematical proof of the reduced energy consumption is missing. Moreover, the energy efficiency is reported to have increased by 59%. In [11], Sun et.al. employs Policy Gradient Reinforcement Learning (PGRL) and improves the power efficiency of NaO by 44%. A critique of [11] reveals that it does not take into account the different states of walking like transition from DSP to SSP, effect of impacts etc. which is critical, when studying energy consumption. Also, the height of CoM is not taken into account in [11], which is important for analyzing the dynamics of the LIPM based humanoid robot like NaO, ASIMO etc.

Besides energy consumption, another major concern in bipedal locomotion is its Stability. The ZMP concept had been heavily relied by researchers for ensuring stability of bipedal robots. Ames et. al [12] used optimization to modify the gait parameters of NaO robot such that the least squares fit of the robotic walking data approximates to the human walking data subjected to the constraints that satisfy partial hybrid zero dynamics. Lin et al. [13] proposed a dynamic balancing methods for bipedal robots using Cerebellar Model Arithmetic Computer (CMAC) Neural Network. The advantage of this method was that, it could fetch optimized gait parameters in real time, according to the terrain profile. Miller et al. [14] developed improved control algorithms for a bipedal robot, that increased its stability. In particular, Miller modelled the gait as a simple oscillator, applied PID control algorithm and then performed neural network training. The advantage of this method was that, without actually knowing the kinematic and dynamic information of the robot, the gait parameters produced, as a result of neural network training, was able to generate stable walking. Zhou et al. [15] used Fuzzy Reinforcement Learning (FRL) to produce stable walking pattern of bipeds. Even though this work did not require the kinematic and dynamic information of the robot, but the demerits of this work comes into picture, when the DoF of the robots increase. With the increase in the DoF of the robots, there is an exponential increase in the number of actions, that has to be taken at every step. That is, it becomes very time consuming to search for every action that suits best for a corresponding state. Jha et al. [16] used GA to form the rule

[1] To understand the difference between the energy consumed, try walking with bent knees for about 5 m. Then, walk back to the starting point normally (knees not bent). The difference between the energy consumed can be clearly understood.

base of the Fuzzy Logic Controller (FLC) for the problem of stable gait generation of bipeds. However, the drawback of this work is that, they did not consider the lateral stability of the robot. Secondly, if the torque (modified by optimized gait generation) exceeds the capacity of the presently equipped motor, then, there may be a need for a little extra force required for maintaining dynamic stability. This extra force needed to maintain stability in bipeds is not considered in this work. Udai proposed an optimum hip trajectory of the robot [17], which had increased the stability of a 10 DoF bipedal robot. In this work, Udai generated an optimum hip location, which would minimize the deviation of the resultant ZMP position from the center of the supporting foot's area. Vundavilli et al. [18, 19] used two hybrid approaches namely GA-NN and GA-FLC to generate stable gaits for bipeds ascending and descending the staircase. GA was used to optimize the weights of NN and the rule base of the FLC offline.

Though the energy and stability functions have been optimized separately as mentioned above, but, it was seen that the two functions behave in a rather contradictory manner [4]. In order to deal with this difficulty, a lot of research has been conducted in the field of legged robots optimizing multiple objectives in the past. Some of them are worth mentioning here. Lee et al. [20] used modified strength pareto-optimality algorithm to generate a set of optimal walking parameters of a bipedal robot, considering three contrasting objectives: energy consumption, walking speed and stability. The energy equation is a function of the height of the swing leg. Furthermore, the stability function defined is a function of hip position, lateral stability and shaking. In order to maximize stability, all the variables mentioned above have to be minimized. The hip joint trajectory is optimized, so that, there is less disturbance in the walking trajectory. The lateral stability is increased by minimizing the error between the calculated and desired ZMP position. The shaking is minimized, so that, there is less sidewise perturbations of the robot during the single support phase. However, the results proposed does not satisfy all the objectives at the same time. Goswami et al. [21] applied multi-objective GA on a 12-DoF biped to optimize two contrasting objectives: walking speed and stability margin. It is seen that, with the increase in walking speed, there is a large oscillation of ZMP position about the foot's geometric center. This, in turn reduces the stability of the robot. In order to increase the stability of the robot, i.e. reducing the ZMP oscillations, the speed of the robot has to be decreased. Pratihar et al. [22] showed that Multi-Objective Particle Swarm Optimization (PSO) performs better than Multi-Objective GA, in finding optimal walking gaits of a bipedal robot. This is because the PSO algorithm searches for optimal solutions both in the local as well as global search space. In this work, the authors dealt with two contrasting objectives: Power consumption and dynamic stability margin. However, the drawback of this research was that, it did not consider lateral stability into account, while defining the stability of the robot. Secondly, the effect of impacts is not reflected on the power consumed by the robot[23, 24]. Thirdly, only one constraint is considered i.e. change in motor torque to be within permissible range, whereas other constraint like motor power rating, motor acceleration limit is not considered. Capi et al. formulated two contrasting objective functions of the bipedal robot namely Minimum Consumed Energy (MCE) and Minimum Change in Torque (MCT). According to Capi [25, 26], the gait generated by optimizing the MCE function produces human-like walking patterns, which consumes less energy. On the other hand, MCT produces more stable motions, due to smooth changes in motor accelerations [27, 28].

Capi et al. used Real Coded Genetic Algorithm (RCGA) to minimize the energy consumed by the bipedal robot. The walk trajectories generated by selecting optimum value of walk parameters resulted in increasing the energy efficiency of the robot by 24%. Jong et al. [8] used GA to minimize energy consumed by a 6-DoF bipedal robot, by selecting optimal locations of the mass centers of the links. However, this method requires one to design a bipedal robot's hardware in such a manner that the locations of CoM of each link corresponds to optimal locations of CoM previously found using GA. Choi et al. [29] used GA to optimize the walking trajectory of IWR-III bipedal robot by minimizing the sum of deviations of velocities, accelerations and jerks in order to maintain continuity of joint trajectories and energy distribution at the via points. But this research does not show how implementing continuity in the joint trajectory results in the improvement of energy consumption [30].

Besides energy consumption, another major concern in bipedal locomotion is its Stability. The concept of ZMP has been heavily relied by researchers for ensuring stability of bipedal robots. In a nutshell, the ZMP concept states that, the CoM projection of a system on the ground should lie within its support polygon. A significant amount of research has been done in order to maintain the posture of the robot in such a manner that its CoM horizontal projection ensures maximum stability. In a nutshell the concept of ZMP states that in order to maintain stability of a bipedal robot, the position of ZMP should lie inside the center of the convex hull formed by the supporting foot. Ames et. al [12, 31] used optimization to modify the gait parameters of NaO robot such that the least squares fit of the robotic walking data approximates to the human walking data subjected to the constraints that satisfy partial hybrid zero dynamics [32].

The novelty of this paper extends to three folds. The concept of orbital energy has been used to develop an energy function which indicates the energy consumed by the robot in one gait cycle. Secondly, this paper proposes a new function which gives a measure of the stability of the bipedal robot not only during the SSP, but also during the transition from the Single Support Phase (SSP) to Double Support Phase (DSP) and vice-versa. The work mentioned in [14], have attempted to build a stability function, which had only taken the stability concern of the robot during the SSP [33, 34]. And finally, this is probably the first paper, which applies Multi Objective Evolutionary Algorithm on NaO humanoid robot. The advantage of such an approach is that, the theoretical results can be verified on a real Nao robot.

2 Assumptions

1. The robot should be flat footed.
2. The foot ground impact energy consumption is not considered to develop the analytical model.
3. Friction forces are not considered for developing the energy and stability functions.
4. The height of the center of mass should vary slightly in the straight line.

3 Stability parameters of Bipedal robots: ZMP and CoM

In this section, We first introduce the notion of two important concepts used throughout this paper i.e ZMP and CoM. The ZMP is defined as a point on the walking surface at which the resultant moment caused by the active forces (gravitational, centrifugal and Coriolis forces) is equal to zero. On the other hand, CoM is defined as a point, where the entire mass of the system is assumed to be concentrated.In order to maintain stability during walking, the ZMP should lie within the support polygon. That means, when the robot is in the Single Support Phase (SSP) the ZMP lies at the center of the foot. On the other hand, when the robot is in the Double Support Phase (DSP), the ZMP lies in the center of the convex hull formed by the two supporting foot In this section, we first introduce the notion of two important concepts used throughout this paper i.e ZMP and CoM. The ZMP is defined as a point on the walking surface at which the resultant moment caused by the active forces (gravitational, centrifugal and Coriolis forces) are equal to zero. On the other hand, CoM is defined as a point, where the entire mass of the system is assumed to be concentrated. It is interesting to know that the position of the ZMP is not directly controllable. This is because, once the foot is in contact with the ground, the reactive forces acting on the ground does not change. However, the position of ZMP can be manipulated indirectly [35] i.e. by varying the position of the CoM of the body. The position of CoM of the robot can be varied by changing the body posture. Another thing worth mentioning here, is the condition ascertaining stability of bipedal robots. In order to maintain stability during walking, the ZMP should lie within the support polygon. This means that when the robot is in the Single Support Phase (SSP) the ZMP lies at the center of the foot. On the other hand, when the robot is in the Double Support Phase (DSP), the ZMP lies in the center of the convex hull formed by the two supporting foot.

3.1 Ideal ZMP and CoM Trajectories

An ideal walk of NaO robot with a fixed step size **B** is shown in the Fig. 1a. The x and y component of ZMP trajectories with respect to time is shown Fig. 1, where **A** is half of the lateral distance between the feet. Mathematically, the horizontal components of the ZMP trajectories can be expressed as:

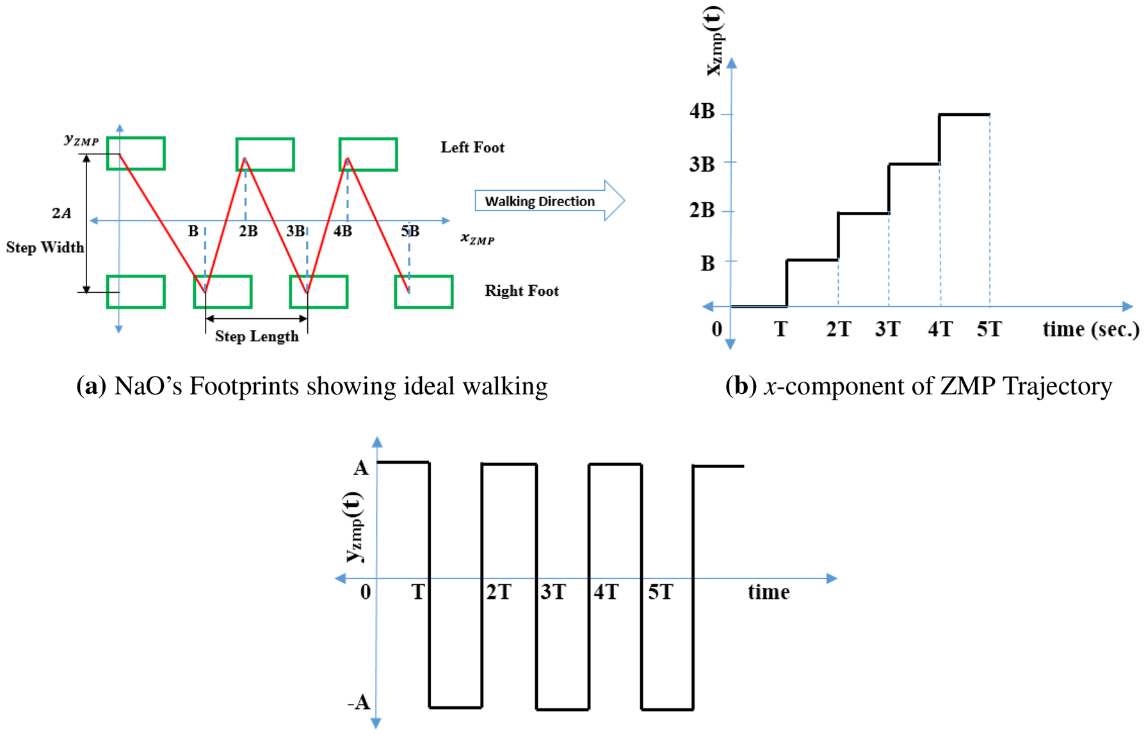
$$x_{zmp}^{ref}(t) = B\sum_{n=1}^{\infty} u(t-nT) \tag{1}$$

$$y_{zmp}^{ref}(t) = Au(t) + 2A\sum_{n=1}^{\infty}(-1)^n u(t-nT) \tag{2}$$

where T is the half period time. The x and y component of ZMP trajectory, shown in Fig. 1b is quite intuitive. The x-component of ZMP is a staircase because, as the robot switches its support from the left leg to the right leg or vice-versa, the CoM effectively increments by a magnitude of the step size (B). Similarly, for the y-component of the ZMP, the upper and the lower edges of the square wave, as shown in Fig. 1c corresponds to the ZMP trajectory in the left and the right leg respectively. However, the above specified trajectories are not realistic. This is because, Eqs. 1 and 2 assume instantaneous exchange of support foot. This assumption can be noticed from the instant change in the ZMP trajectory in Fig. 1b, c.

3.2 Realistic/practical ZMP-CoM Trajectories

Ideally, the ZMP shifts instantly from the support polygon of left foot to the support polygon of the right foot, during the transition from the Double Support Phase (DSP) to Single Support Phase (SSP). But in a practical case, a slight modification of the ZMP trajectory was suggested by [36]. Instead of the ZMP shifting instantly, the ZMP follows the CoM during the transition from the DSP to SSP as shown in the Fig. 2.



(a) NaO’s Footprints showing ideal walking

(b) x-component of ZMP Trajectory

(c) y-component of ZMP Trajectory

Fig. 1 Ideal NaO Robot’s Walk and ZMP Trajectories

The X-directional ZMP trajectory is given by:

$$x_{zmp}(t) = \begin{cases} (k_x/t_s)t, & if \quad 0 \le t \le t_s \\ B, & if \quad t_s < t < T - t_s \\ (2B - k_x) + \dfrac{k_x}{t_s}(t - (T - t_s)), & if \quad T - t_s \le t \le T \end{cases} \tag{3}$$

where B is the step length, ts is the transition time required to change from DSP to SSP and T is the half period time, required to complete one step. Mathematically, T represents the time required by the biped to switch from DSP to SSP and then again back to DSP.k_x is the slope of the line Fig. 2a. During the DSP, i.e. $(0 \le t \le t_s)$ and $(T - t_s \le t \le T)$, the CoM trajectory follows the ZMP trajectory.

$$x_{cm}(t) \triangleq x_{zmp}(t) \tag{4}$$

During the SSP, i.e. $(t_s < t < T - t_s)$ the CoM trajectory is defined by the equation:

$$\ddot{x}_{cm} - \omega^2 x_{cm} = -\omega^2 B \tag{5}$$

where $\omega = \sqrt{\frac{g}{z_{cm}}}$. The solution of Eq. 5 is given by:

$$x_{cm}(t) = C_1 \cosh(\omega(t - t_s)) + C_2 \sinh(\omega(t - t_s)) + B \tag{6}$$

Using the initial and the final conditions, the values of C_1, C_2 and, k_x are:

$$C_1 = k_x - B; \quad C_2 = k_x/\omega t_s; \quad k_x = \frac{B t_s \omega}{\omega t_s + tanh\left(\omega\left(\frac{T}{2} - t_s\right)\right)}$$

The Y-directional ZMP trajectory is given by:

$$y_{zmp}(t) = \begin{cases} (k_y/t_s)t, & if \quad 0 \le t \le t_s \\ A, & if \quad t_s < t < T - t_s \\ (k_y/t_s)(T - t), & if \quad T - t_s \le t \le T \end{cases} \tag{7}$$

where A is the distance between the left foot and the right foot. During the DSP, i.e. $(0 \le t \le t_s)$ and $(T - t_s \le t \le T)$, the CoM trajectory follows the ZMP trajectory.k_y is the slope of the line Fig. 2b.

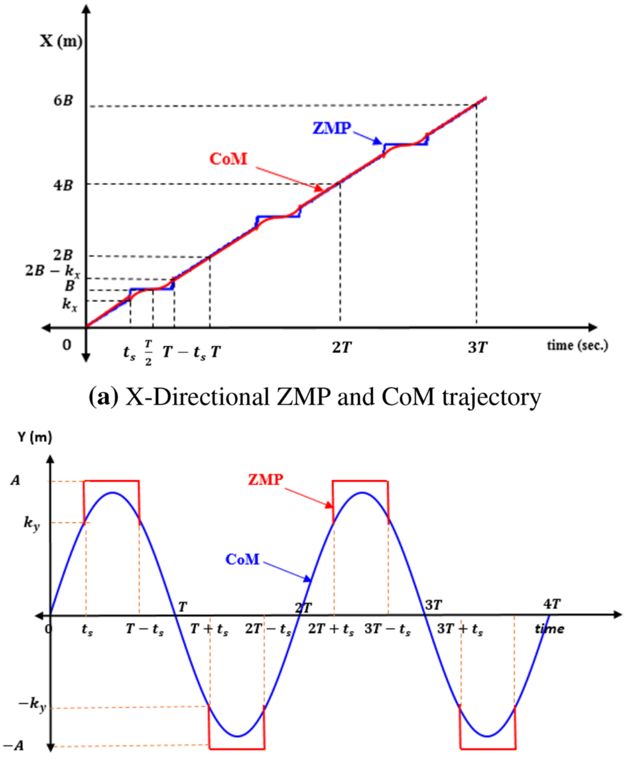
$$y_{cm}(t) \triangleq y_{zmp}(t) \tag{8}$$

During the SSP, i.e. $(t_s < t < T - t_s)$ the CoM trajectory is defined by the equation:

$$\ddot{y}_{cm} - \omega^2 y_{cm} = -\omega^2 A \tag{9}$$

The solution of Eq. 9 is given by:

$$y_{cm}(t) = C_3 \cosh(\omega(t - t_s)) + C_4 \sinh(\omega(t - t_s)) + A \tag{10}$$



(a) X-Directional ZMP and CoM trajectory

(b) Y-Directional ZMP and CoM trajectory

Fig. 2 Practical ZMP and CoM trajectories

Using the initial and the final conditions, the values of C_3, C_4 and, k_y are:

$$C_3 = k_y - A \quad ; \quad C_4 = k_y/\omega t_s \quad ; \quad k_y = \frac{At_s\omega \tanh\left(\omega\left(\frac{T}{2} - t_s\right)\right)}{1 + \omega t_s + \tanh\left(\omega\left(\frac{T}{2} - t_s\right)\right)}$$

However, Eqs. 6 and 10 involve hyperbolic terms, which are unbounded and are difficult to be realized. Therefore an approximate solution is formulated by expressing the ZMP trajectories by Fourier series [37]. Since $y_{cm}^{ref}(t)$ is a periodic function its Fourier series expansion is given by:

$$y_{cm}(t) = \sum_{n=1}^{n=\infty} \frac{2AT^2\omega^2(1 - \cos n\pi)}{n\pi(T^2\omega^2 + n^2\pi^2)} \sin\left(\frac{n\pi}{T}t\right) \tag{11}$$

The x-component of ZMP trajectory(shown in Fig. 1b) is not periodic in nature, and its Fourier series cannot be found. An approximate analysis was given in [37] to calculate the Fourier series of $x_{cm}^{ref}(t)$ as:

$$x_{cm}(t) = \frac{B}{T}\left(t - \frac{T}{2}\right) + \sum_{n=1}^{n=\infty} \frac{BT^2\omega^2(1 + \cos n\pi)}{n\pi(T^2\omega^2 + n^2\pi^2)} \sin\left(\frac{n\pi}{T}t\right) \tag{12}$$

Once the ZMP and CoM trajectories are formed, we use them to build up the energy and stability equation at different support phases.

4 Formulation of the objective function

4.1 The energy function

Mostly, the energy functions derived for a bipedal robot uses the concept of the change in torque at the joints [4, 6]. In this work, the authors have developed an Energy function using the concept of Orbital Energy. The derivation of Energy function is valid for NaO robot because, the NaO robot's hip trajectory (representing CoM) follows a straight line while walking, maintaining a constant height above the ground (the only major requirement for the application of the Orbital Energy Concept).

A generalized equation can be written as:

$$E_{x(i)}(t) = -\frac{g}{2z_{cm}}\left(x_{cm(i)} + y_{cm(i)}\right)^2 + \frac{1}{2}\left(\dot{x}_{cm(i)} + \dot{y}_{cm(i)}\right)^2 \tag{13}$$

$$E_{y(i)}(t) = -\frac{g}{2z_{cm}}\left(-x_{cm(i)} + y_{cm(i)}\right)^2 + \frac{1}{2}\left(-\dot{x}_{cm(i)} + \dot{y}_{cm(i)}\right)^2 \tag{14}$$

where

$$i = \begin{cases} 1, \; if & 0 \le t \le t_s \\ 2, \; if & t_s < t < T - t_s \\ 3, \; if & T - t_s \le t \le T \end{cases}$$

The robot was made to walk for one complete gait cycle. The walking cycle was divided into three different phases : DSP, SSP and DSP.

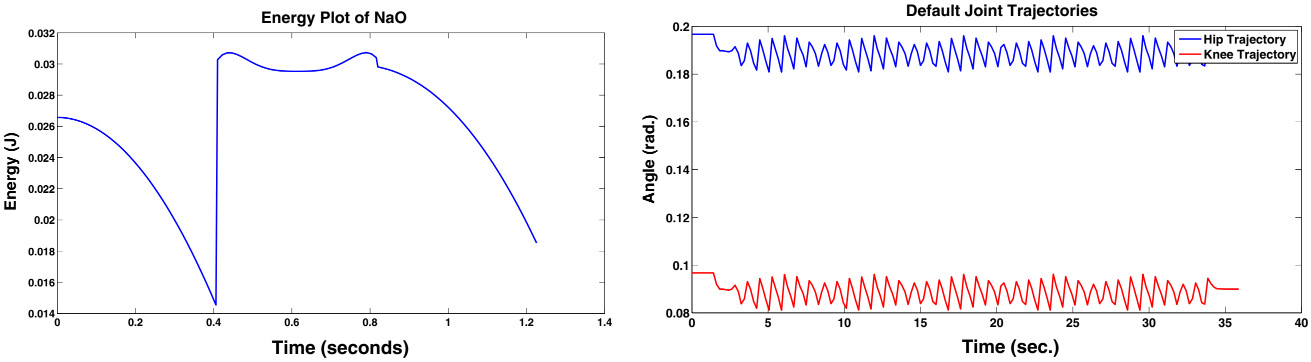
Case 1: $0 \le t \le t_s$

During this time interval, the robot is in DSP. The CoM and ZMP trajectories are almost similar which is given by Eqs. 4 and 8. The equation for orbital energy can therefore be obtained by substituting the values of x_{cm} and y_{cm} in the Eqs. 37 and 38 as:

$$x_{cm} = \frac{k_x}{t_d}t; \quad y_{cm} = \frac{k_y}{t_d}t \tag{15}$$

Substituting $x_{cm(1)} = x_{cm}$ and $y_{cm(1)} = y_{cm}$ from equation and $\dot{x}_{cm(1)} = \frac{\partial}{\partial t}x_{cm(1)}$ and $\dot{y}_{cm(1)} = \frac{\partial}{\partial t}y_{cm(1)}$ in Eqs. 37 and 38, E_{x1} and E_{y1} can be calculated. E_1 i.e. the energy during this interval, is then computed as:

$$E_1(t) = \sqrt{(E_{x1}(t))^2 + (E_{y1}(t))^2} \tag{16}$$



(a) Energy Plot using Default Aldebaran Walk parameters

(b) Default Hip and Knee Trajectories of NaO

Fig. 3 Energy plot and joint trajectories of default walk

$$E_1 = \int_{t=0}^{t=t_s} E_1(t)dt \tag{17}$$

Case 2: $t_s \leq t \leq T - t_s$

During this time interval, the robot is in SSP. The CoM and ZMP trajectories are different and is given by

$$x_{zmp}(t) = B \quad y_{zmp}(t) = A$$

The equation for orbital energy during this time interval can be obtained by substituting the values of x_{cm} and y_{cm} mentioned above in the Eqs. 37 and 38.

$$x_{cm}(t) = \frac{B}{T}\left(t - \frac{T}{2}\right) + \sum_{n=1}^{n=\infty} \frac{BT^2\omega^2(1 + \cos n\pi)}{n\pi(T^2\omega^2 + n^2\pi^2)} \sin\left(\frac{n\pi}{T}t\right) \tag{18}$$

$$y_{cm}(t) = \sum_{n=1}^{n=\infty} \frac{2AT^2\omega^2(1 - \cos n\pi)}{n\pi(T^2\omega^2 + n^2\pi^2)} \sin\left(\frac{n\pi}{T}t\right) \tag{19}$$

Substituting $x_{cm(2)} = x_{cm}$ and $y_{cm(2)} = y_{cm}$ from the Eqs. 18 and 19 and $\dot{x}_{cm(2)} = \frac{\partial}{\partial t}x_{cm(2)}$ and $\dot{y}_{cm(2)} = \frac{\partial}{\partial t}y_{cm(2)}$, we obtain E_{x2} and E_{y2}. E_2 is then computed as:

$$E_2(t) = \sqrt{(E_{x2}(t))^2 + (E_{y2}(t))^2} \tag{20}$$

$$E_2 = \int_{t=t_s}^{t=T-t_s} E_2(t)dt \tag{21}$$

Case 3: $T - t_s \leq t \leq T$

During this time interval, the robot is again in DSP, thus completing one complete gait cycle.The CoM and ZMP trajectories are almost same as mentioned in Case 1. However, the values of x_{cm} and y_{cm} changes as shown in Fig. 2, and is given by:

$$x_{cm}(t) = (2B - k_x) + \left(\frac{k_x}{t_s}\right)(t - (T - t_s)) \triangleq x_{zmp}(t) \tag{22}$$

$$y_{cm}(t) = \left(\frac{k_y}{t_s}\right)(T - t) \triangleq y_{zmp}(t) \tag{23}$$

Substituting $x_{cm(3)} = x_{cm}$ and $y_{cm(3)} = y_{cm}$ from the Eqs. 22 and 23 and $\dot{x}_{cm(3)} = \frac{\partial}{\partial t}x_{cm(3)}$ and $\dot{y}_{cm(3)} = \frac{\partial}{\partial t}y_{cm(3)}$, we obtain E_{x3} and E_{y3}. E_3 is then computed as:

$$E_3(t) = \sqrt{(E_{x3}(t))^2 + (E_{y3}(t))^2} \tag{24}$$

$$E_3 = \int_{t=T-t_s}^{t=T} E_3(t)dt \tag{25}$$

From the above discussions, it can be concluded that energy is a function of x_{zmp}, y_{zmp}, z_{cm} and t_s. The co-ordinates of CoM and ZMP, in turn depends on the value of walk parameters step length B and lateral foot distance B.

The Energy function therefore, is:

$$\boxed{E(t_s, z_{cm}, A, B) = \frac{1}{3}\sum_{j=1}^{j=3} E_j} \tag{26}$$

The aim is to minimize the energy consumed, given by Eqs. 26, subjected to the constraints specified by Aldebaran Robotics (manufacturer of NaO robot):

$$0.14 \text{ m} < z_{cm} < 0.33 \text{ m}; \quad 0.4 \text{ s.} \leq t_s \leq 0.6 \text{ s.}$$

$$0.101 \text{ m} \leq A \leq 0.160 \text{ m}; \quad 0.01 \text{ m} \leq B \leq 0.08 \text{ m}$$

4.1.1 Optimization of the energy function: results

RCGA is used to optimize the energy function given by Eqs. 26, subjected to the constraints to the walk parameters mentioned above. This section shows the default as well as the optimized energy plots of the NaO robot. Corresponding to the optimized walk parameters, the energy

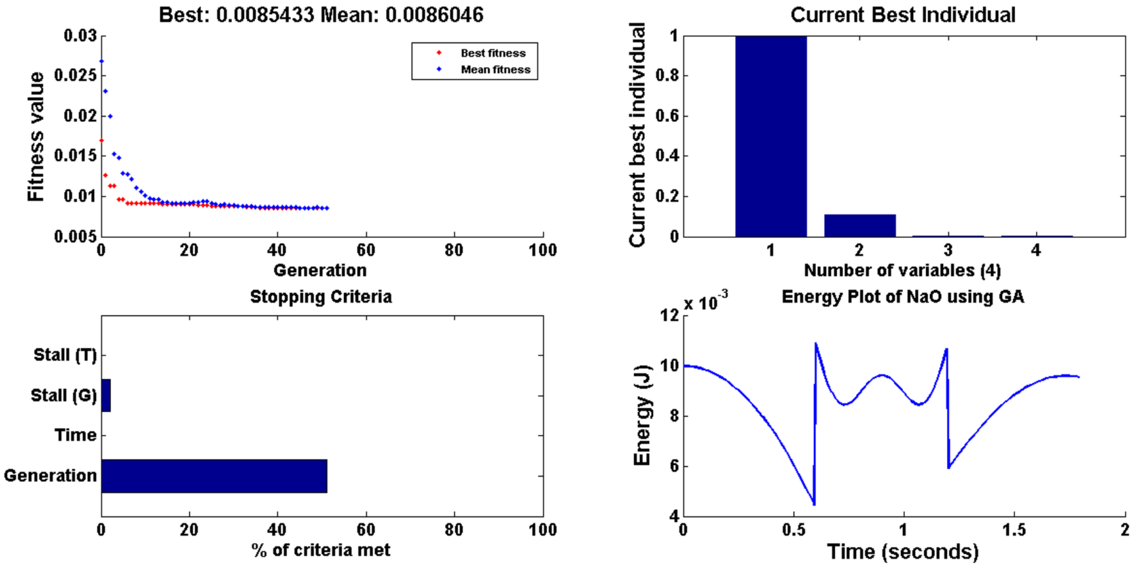
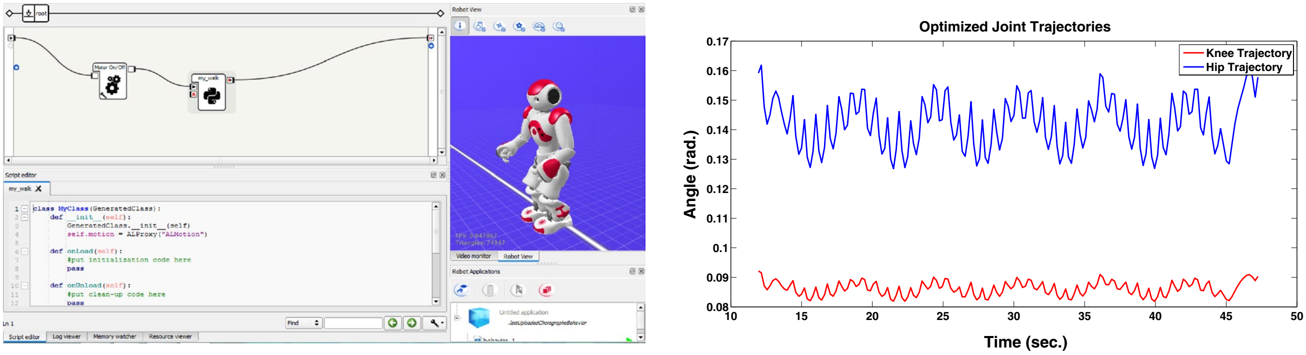


Fig. 4 Energy optimization using GA



```
class MyClass(GeneratedClass):
    def __init__(self):
        GeneratedClass.__init__(self)
        self.motion = ALProxy("ALMotion")

    def onLoad(self):
        #put initialization code here
        pass

    def onUnload(self):
        #put clean-up code here
        pass
```

(a) NaO Simulation in Choregraphe

(b) Optimized Hip and Knee Trajectories of NaO

Fig. 5 Simulation of NaO's walking in choregraphe and optimized walk Trajectories

plot obtained is also compared with the PGRL and stiffness based approach for calculating the energy of the robot. Finally, the joint trajectories of the hip and knee joint of the bipedal robot, ensuring minimum energy consumption is generated.

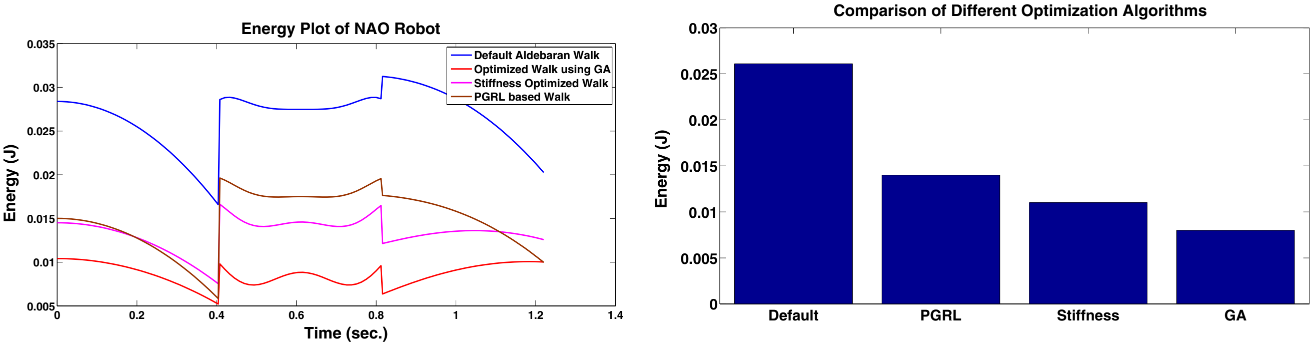
The default Aldebaran walk engine [38] had the values of input parameters set as:

$$t_s = 0.41 \text{ s} \quad z_{cm} = 0.25 \text{ m} \quad A = 0.135 \text{ m} \quad B = 0.04 \text{ m}$$

The energy plot of the default Aldebaran walk over a time period T is shown in the Fig. 3a. Using Eq. 26, the energy consumed by the robot during walking was 0.0261 J.

Figure 3a shows that the energy consumed by the robot initially is high. A discontinuity in the energy graph is seen at time $t = t_s = 0.41$ s, when the robot transitions from DSP to SSP. During the SSP, i.e. $0.41 < t < 0.82$, the energy function is sinusoidal in nature. This is because, the COM trajectory given by Eqs. 11 and 12 includes sinusoidal terms. For simplicity, we have taken harmonics only upto the second order. If the harmonics are increased, the number of peaks appearing during this interval in the energy graph will also increase subsequently. At t = 0.82 ($T - t_s$), the robot transitions to DSP. During this interval, the energy graph is similar to the DSP, which was encountered at the start of the gait cycle. Figure 3b shows the hip and the knee trajectories of a real NaO robot using default walk parameters.

The objective function was then optimized by using GA. The results obtained by using GA is shown in the Fig. 4. The top left plot shows how the fitness function (objective function) gradually converges to a global minimum value. The best fitness value obtained using GA is 0.00854 J. The optimized value of the four walk parameters are shown in the top right figure. Due to different constraints of walk



(a) Energy Plot of NaO using different optimization algorithms

(b) Comparison of energy consumption over one gait cycle

Fig. 6 Comparison of different optimization algorithms

parameters, the value of walk parameters are normalized, such that they lie between the given constraints. For example, $t_s = 1$ shown in the Fig. 4 means t_s is at the upper extrema of the constraint for t_s ($t_s = 0.6$). By using GA, at the end of 51 iterations, the optimized value of the parameters are:

$$t_s = 0.6\ \text{s} \quad z_{cm} = 0.161\ \text{m} \quad A = 0.101\ \text{m} \quad B = 0.01\ \text{m}$$

The energy plot of the optimized walk is depicted from the bottom right figure. When the above optimized walk parameters was substituted in Eq. 26, the energy consumed by the robot was 0.0085 J. This marked an increased power efficiency of NaO robot by 69.35%. This is an improvement over results reported in [10] and [11] which proposes increased power efficiency by 44 and 59%, respectively.

The simulations were carried out in Choregraphe software as shown in Fig. 5a. The optimized walk parameters were tested on a real NaO robot and the corresponding hip and knee trajectories were generated as shown in Fig. 5b. It can be clearly noticed from Figs. 3b and 5b that, the hip trajectory is close to z_{cm}. Also, the optimized joint trajectory has more peak to peak deviations as compared to the the default joint trajectories. This is because, the value of the walk parameter t_s of a NaO robot cannot be directly controlled.

Instead, t_s can be controlled by adjusting the value of *step height* (h) walk parameter. The default hip trajectory has less deviations since, step height is set to minimum ($h = 0.005m$). On the other hand, the optimized hip trajectory has more deviations since the step height was increased to $h = 0.04$ m, thereby increasing the value of t_s.

Figure 6a shows the energy plot obtained by setting the optimized value of walk parameters obtained from different algorithms like PGRL, Stiffness based walking, RCGA, and default Aldebaran walking. Figure 6b shows that when NaO robot was made to walk for one gait cycle, the energy consumed was 0.0261 J. The stiffness of the motor joints were varied in [10] and the energy consumed is reported to be 0.011 J. On applying optimization techniques like PGRL [11], the energy consumed by NaO robot was reported to be 0.014 J. By applying RCGA to our energy function, the energy consumed by the robot was 0.0085 J. This marks an increase in energy efficiency of the NaO robot by 69.35% over default Aldebaran walk, 15.35% over PGRL based walk and, 10.35% over Stiffness based walk.

4.2 The stability function

The ZMP position must be at the center of support polygon of the bipedal robot, in order to ascertain maximum stability. Recent works on formulating a stability function [6] had used the concept of ZMP deviation from the mean position. That is, the Stability function indicates the measure of deviation of the *calculated* ZMP position (calculated from the real robot's dynamics) from the desired ZMP position (at the center of the support polygon). However, the stability functions proposed earlier, requires the ZMP trajectory to be at the center of the robot's foot only during the SSP. Since, it is well known fact that a normal gait cycle consists of alternate phases of DSP and SSP. Therefore, the consideration of DSP in designing the stability function plays an equally important role. In this work, the author has considered both the support phases of walking, i.e. DSP and SSP. The main idea behind the proposal of this novel stability function is that, the ZMP trajectory remains in the center of the support polygon not only during the SSP, but also during the DSP.

4.2.1 *Derivation of the stability function*

The Stability function (ζ) is given by the difference of the $y-$component[2] of calculated ZMP position ZMP_{calc} given by

[2] The x-component of ZMP trajectory does not account much for ensuring the stability of the bipedal robots. This is because, the x-component of ZMP gives information only about the distance travelled by the bipedal robot and depicts nothing about its stability.

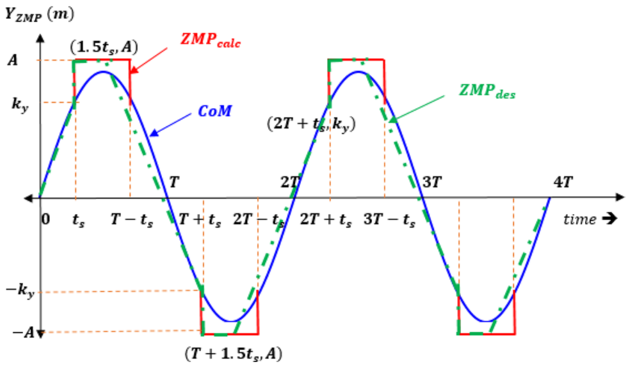


Fig. 7 CoM, calculated ZMP and desired ZMP Trajectories of a Bipedal robot

Eq. 7 (explained in Sect. 2.2) and the desired ZMP_{des} (explained in this section). Mathematically, this can be represented as:

$$\zeta = ||ZMP_{calc} - ZMP_{des}|| \tag{27}$$

Lower values of ζ indicates that the robot is highly stable, ensuring that the ZMP trajectory remanins at the center of the support polygon in both DSP as well as SSP. Higher values of ζ states that there is a large deviation of the calculated ZMP from the desired ZMP, indicating that the robot is unstable. The green dashed line in Fig. 7 shows the desired ZMP trajectory. The red solid line indicates the calculated ZMP trajectory which is same as the one explained earlier. The blue line represents the CoM trajectory of the bipedal robot. Since ZMP_{calc} is defined by Eq. 7, the only unknown in Eq. 27 is ZMP_{des}. Similar to the Energy function derived in Sect. 3.1, the ZMP trajectory is modified and its equation is formulated in different time intervals.

Case 1: $0 \le t \le t_s$

During this interval, the bipedal robot is in transition from DSP to SSP. The ZMP trajectory (indicated by green dashed line) is parallel to the ZMP trajectory obtained during the interval $T + 1.5t_s \le t \le 2T + t_s$. Thus,

$$y^{zmp}_{des}(t) = \frac{2}{5t_s}\left(A + k_y\right)t + c$$

At $t = t_s$, $y^{zmp}_{des} = k_y$. With these conditions, solving for constant c, the equations for desired ZMP trajectory can be written as:

$$y^{zmp}_{des}(t) = \frac{2}{5t_s}\left(A + k_y\right)t - \frac{2}{5}A + \frac{3}{5}k_y \tag{28}$$

During this interval, the equation for y^{zmp}_{calc} is given by equation:

$$y^{zmp}_{calc}(t) = \frac{k_y}{t_s}t \tag{29}$$

Case 2: $t_s \le t \le 1.5t_s$

During this time interval, the desired ZMP trajectory is same as the calculated ZMP trajectory. The ZMP position shifts from the heel towards the center of the foot. That is:

$$y^{zmp}_{des}(t) = A \tag{30}$$

During this interval, the equation for y^{zmp}_{calc} is given by equation:

$$y^{zmp}_{calc}(t) = A \tag{31}$$

Case 3: $1.5t_s \le t \le T + t_s$

Instead of the ZMP shifting towards the toe, during this interval, the ZMP trajectory is a straight line which is parallel to the sides of the support polygon formed by the left and the right foot. The slope of the dashed line (green) is given by:

$$m = \frac{A + k_y}{0.5t_s + T} = -\frac{2}{5t_s}(A + k_y) \quad (\because T = 3t_s)$$

Thus, the equation of this line can be written as:

$$y^{zmp}_{des}(t) = -\frac{2}{5t_s}(A + k_y)t + c$$

From the graph shown in Fig. 7, at time $t = 1.5t_s \Rightarrow y^{zmp}_{des}(t) = A$. Thus, solving for the unknown the desired ZMP equation during this interval is:

$$y^{zmp}_{des}(t) = -\frac{2}{5t_s}(A + k_y)t + \frac{8}{5}A + \frac{3}{5}k_y \tag{32}$$

During this interval, the equation for y^{zmp}_{calc} is splitted in 2 time intervals i.e. $1.5t_s \le t \le T - t_s$ and $T - t_s \le t \le T$: During the interval, $1.5t_s \le t \le T - t_s$, the equation for y^{zmp}_{calc} is :

$$y^{zmp}_{calc}(t) = \frac{k_y}{t_s}t \tag{33}$$

During the interval, $T - t_s \le t \le T$, the equation for y^{zmp}_{calc} is :

$$y^{zmp}_{calc}(t) = \frac{k_y}{t_s}(T - t) \tag{34}$$

Combining Eqs. 28, 30 and 32 into one single piece-wise equations, the equation for y^{ZMP}_{des} can be obtained as:

$$y^{zmp}_{des}(t) = \begin{cases} \frac{2}{5t_s}\left(A + k_y\right)t - \frac{2}{5}A + \frac{3}{5}k_y, & if \quad 0 \le t \le t_s \\ A, & if \quad t_s < t < 1.5t_s \\ -\frac{2}{5t_s}(A + k_y)t + \frac{8}{5}A + \frac{3}{5}k_y, & if \quad 1.5t_s \le t \le T \end{cases} \tag{35}$$

Once the equation for $y^{zmp}_{des}(t)$ is formulated, the Stability function can be obtained from the Eq. 27 as:

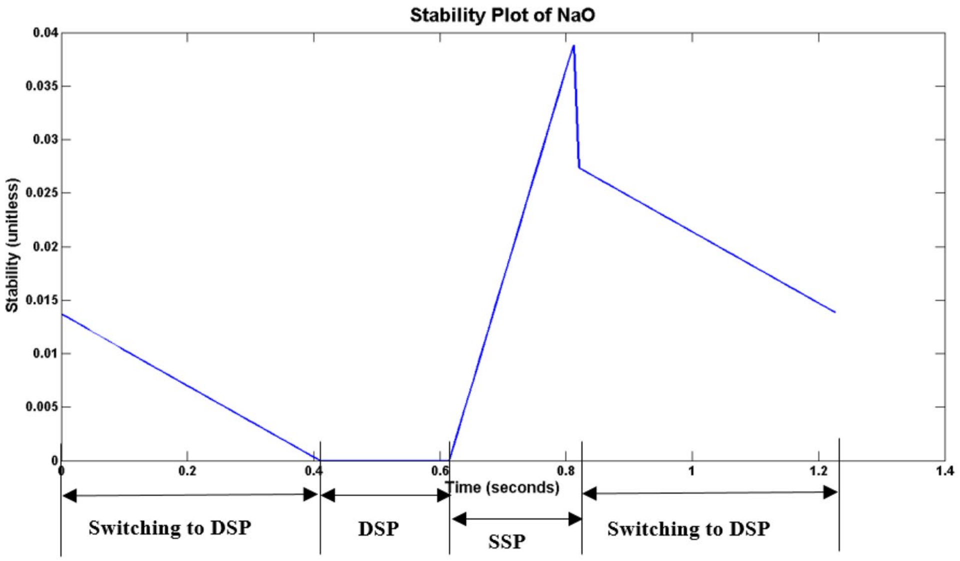


Fig. 8 Stability plot using default Aldebaran walk parameters

$$\zeta = \begin{cases} \frac{3k_y}{5t_s}t - \frac{2A}{5t_s}t + \frac{2}{5}A - \frac{3}{5}k_y, & if \quad 0 \le t \le t_s \\ 0, & if \quad t_s \le t \le 1.5t_s \\ \frac{2}{5t_s}(A + k_y)t - \frac{3}{5}A - \frac{3}{5}k_y, & if \quad 1.5t_s < t < T - t_s \\ -\frac{k_y}{t_s}T - \frac{3k_y}{5t_s}t + \frac{2A}{5t_s}t - \frac{8}{5}A - \frac{3}{5}k_y, & if \quad T - t_s \le t \le T \end{cases} \tag{36}$$

Once, the Stability function is formulated, the aim is to minimize the average value of ζ over the gait cycle T subjected to the constraints on the walk parameters specified by Aldebaran Robotics (manufacturer of NaO) as:

$$0.14\text{ m} < z_{cm} < 0.33\text{ m}; \quad 0.4\text{ s.} \le t_s \le 0.6\text{ s.}$$

$$0.101\text{ m} \le A \le 0.160\text{ m}; \quad 0.01\text{ m} \le B \le 0.08\text{ m}$$

The minimization of ζ means that the ZMP trajectory remains at the center of the support polygon throughout the walking cycle. In order to realize the ZMP trajectory given by Eq. 35 on a real NaO robot, the CoM equation needs to computed by inputting the ZMP_{des} equations into Eq. 11.

4.2.2 Optimization of the stability function: results

The Stability function given by Eq. 36 is optimized using RCGA subjected to the constraints on the walk parameters mentioned above. This section illustrates the default as well as the optimized stability plots of the NaO robot.

The default Aldebaran walk engine [38] had the values of the input parameters of NaO robot set as:

$$t_s = 0.41\text{ s} \quad z_{cm} = 0.25\text{ m} \quad A = 0.135\text{ m} \quad B = 0.04\text{ m}$$

The Stability plot of the default Aldebaran walk over one gait cycle is shown in Fig. 8. Using Eq. 36, the average stability of the bipedal robot over one gait cycle was measured to be 0.0124. The numerical value just stated, indicates the average degree of deviation of the ZMP_{des} from the ZMP_{calc} trajectory.

From Fig. 8, it can be inferred that, when the robot is switching to DSP, the stability of the robot increases. This can be seen from the decreasing value of ζ during the interval $0 \le t \le 0.41$. The robot comes to complete DSP during the interval $0.41 \le t \le 0.6$. During this time interval, the robot possesses maximum stability ($\zeta = 0$). In the next time interval $0.6 \le t \le 0.82$, the robot switches to SSP. The SSP is a highly unstable state. This can be seen from the sudden rising value of ζ as shown in the Fig. 8. During the time interval, $0.82 \le t \le 1.25$, the robot is again switching to DSP. The stability of the robot increases during this time interval. This is indicated by the decreasing value of ζ, as shown in the Fig. 8.

The stability function was then optimized using RCGA. The results obtained by using GA is shown in Fig. 9. The top left plot shows how the fitness function (minimizing the stability function) gradually converges to a global minimum value. The best fitness value obtained using GA is 0.0028. The optimized value of the walk parameters is shown by means of a bar graph in the top right plot. It is seen that, by the end of 51 iterations, the optimized value of the walk parameters are:

$$t_s = 0.6\text{ s} \quad z_{cm} = 0.161\text{ m} \quad A = 0.101\text{ m} \quad B = 0.01\text{ m}$$

The stability plot of the optimized walk is depicted from the bottom right figure. Substituting the above

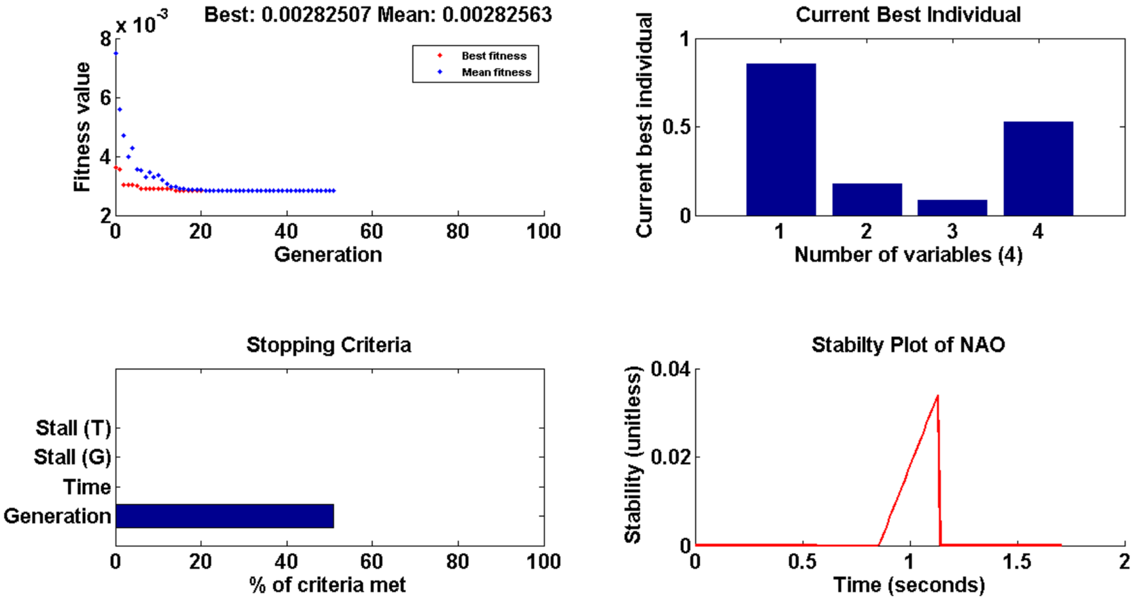


Fig. 9 Optimization of stability function using RCGA

Table 1 Energy optimized walk parameters applied on energy and stability functions

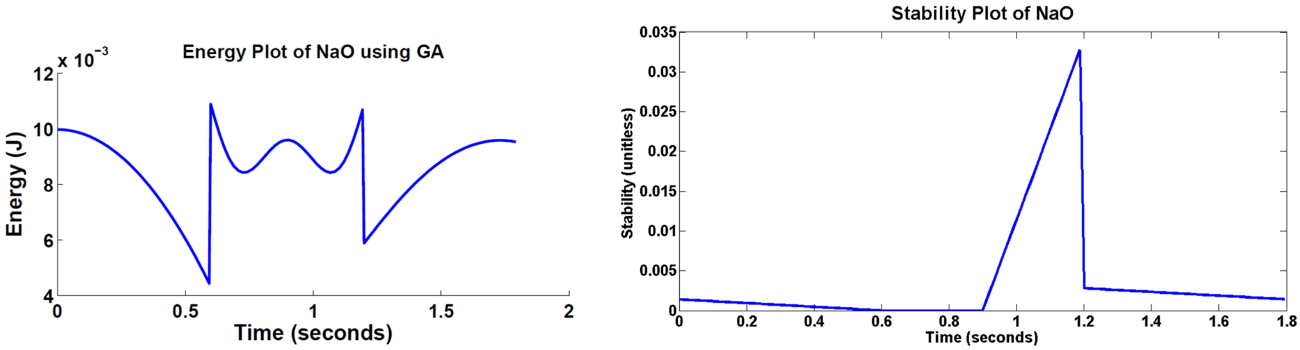
	t_s	z_{cm}	A	B	Energy (J)	Stability
Default	0.41	0.25	0.135	0.04	0.0261	0.0124
Optimized	0.6	0.161	0.101	0.01	0.085	0.0037

walk parameters in the Eq. 36, the value of ζ obtained, is 0.0028. This marks an increase in the stability of the bipedal robot by 77.42% over the existing ζ value obtained using default walk parameters. The Stability plot of NaO, shown in Fig. 9, shows that the robot is highly stable during its transition from DSP to SSP. The robot becomes unstable during SSP, which is quite normal. It is observed that, with the implications of the optimized walk parameters on the robot, the stability of the biped increases. This is due to the sudden decrease in the value of ζ during the transition of DSP to SSP and vice-versa as shown in Fig. 9.

5 Multi-objective optimization problem

The use of the optimization was greatly felt during World War I, when the demand for the number of products (weapons, processed food etc.) exceeded the production capacity of industries. Due to ongoing war, resulting in time crisis, new factories could not be set up to meet the demands of the public. This gave rise to a method called Optimization, first introduced by Fermat and Lagrange, where the existing facilities of the factories were used wisely (by altering the value of input parameters) so as to maximize the yield using the pre-existing state of the art. Optimization may be broadly classified into two categories based on the number of objectives to be optimized. They are: (1) Single Objective Optimization, (2) Multi-Objective Optimization.

1. *Single objective optimization* In this method, there is only one objective function which needs to be optimized. For instance, both the functions (Energy and Stability) derived in Sect. 3, were optimized separately. That is, when the Energy function given by Eq. 26 was optimized using RCGA, it yielded different set of optimum walk parameters as compared to those obtained from the optimization of the Stability function given by Eq. 36. To illustrate the effect of Single Objective Optimization, consider Table 1, where the optimized walk parameters obtained for Energy function is applied to the Stability function. Table 1 shows the default as well as the optimized walk parameters obtained from the optimization of the Energy function. The optimized walk parameters were substituted in the Stability function and the resulting value of the stability is noted. Figure 10a shows the optimized energy plot of NaO robot. By using the optimized walk parameters (obtained by applying RCGA on the Energy function), the Stability plot of NaO robot is



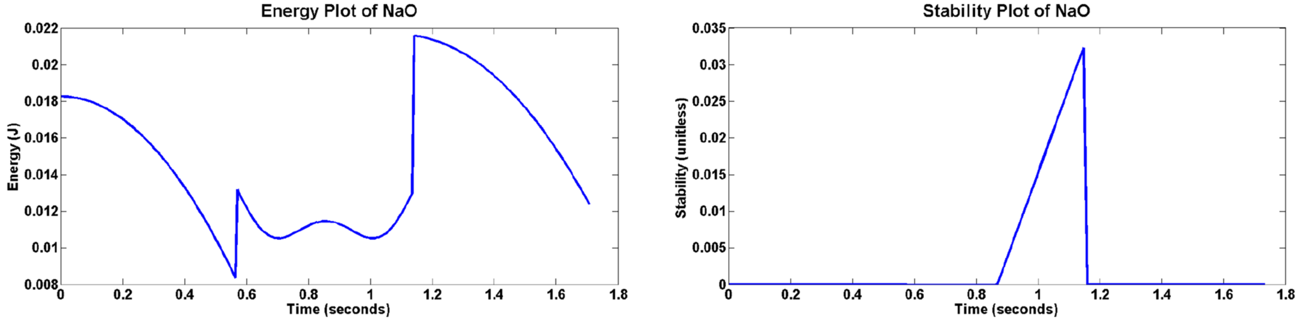
(a) Optimization of the Energy Function

(b) Stability Plot using the Energy optimized walk parameters

Fig. 10 Effect of energy optimized walk parameters on energy and stability functions

Table 2 Stability optimized walk parameters applied on energy and stability functions

	t_s	z_{cm}	A	B	Energy (J)	Stability
Default	0.41	0.25	0.135	0.04	0.0261	0.0124
Optimized	0.5761	0.1732	0.1059	0.0469	0.0148	0.0028



(a) Energy Plot using the Stability optimized walk parameters

(b) Optimization of Stability function

Fig. 11 Effect of stability optimized walk parameters on energy and stability functions

generated. It can be noted from Table 1 that, the optimized walk parameters gives a minimum energy consumption of 0.0085 J, but, the same parameters yield Stability value as 0.0037. The Stability function was optimized in Sect. 3.2, whose optimum value came to be 0.0028. Thus, it can be seen here that, only one objective could be satisfied. To further exemplify the meaning of Single Objective Optimization, consider the reverse case, where the stability optimized walk parameters are applied to the Energy function. Table 2 shows the value of energy consumed by the robot, differs from the optimized energy value shown in Fig. 1, when stability optimized walk parameters are applied on energy equation given by Eq. 26. Figure 11a shows the Energy plot obtained for a bipedal robot using stability optimized walk parameters. It can be clearly seen from Table 2, that the energy function does not yield optimum value (E=0.0085 J), when the stability optimized walk parameters are applied on the energy equation. On the other hand, the stability function has been optimized here, which yields a minimum value of 0.0028. Thus, from the above discussions, it is clear that the energy and stability of the robot can be best optimized when they are handled singly. This concept of applying optimization separately on different objective functions is known as Single Objective Optimization.

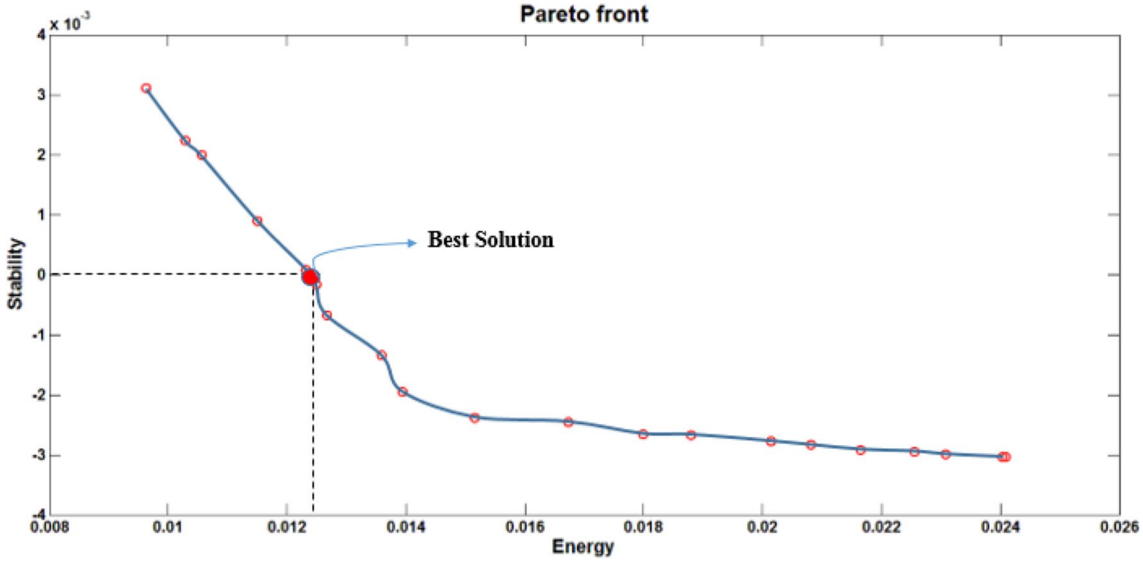


Fig. 12 Pareto optimal front

Table 3 Pareto optimal solutions of the multi-objective optimization problem

Weights		Walk parameters				Fitness value		% Efficiency ↑ in	
$w_1(E = f_1(x))$	$w_2(\zeta = f_2(x))$	t_s	z_{cm}	A	B	Energy (J)	Stability	Energy	Stability
0.1	0.9	0.401	0.3216	0.1398	0.0252	0.0241	−0.0027	7.66	78.22
0.2	0.8	0.4015	0.3215	0.11	0.021	0.0152	−0.0025	41.76	79.83
0.4	0.6	0.417	0.3168	0.1082	0.0165	0.0139	−0.0019	46.74	84.67
0.8	0.2	0.5938	0.1687	0.1045	0.0144	0.0086	0.0031	67.05	75

2. *Multi-objective optimization* In this kind of optimization problems, there are more than one objective functions. Consider for instance the problem of stability and energy consumption of the bipedal robot. Since both the functions (energy and stability) have to be applied on the same bipedal robot, therefore the user needs a single set of walk parameters that can optimize both energy consumption as well as the stability of the robot at the same time. However, it was seen in the previous section that, the optimization of one function leads to worsening of the other. In particular, when the energy function of the robot was optimized, the stability of the robot decreased (high value of ζ). Also, the optimization of the stability function, ended up in greater energy consumption by the robot. Thus, the two functions are clearly contrasting in nature. The drawback in such cases is that, there is no single best solution. To encounter such problems, a set of such optimal solutions are found, each satisfying the desired objectives to a certain degree. The problem of handling this problem is discussed in detail in Sect. 4.1.

5.1 Solution to the problem: energy Vs. stability

Multiple objective optimization (minimizing the energy consumption as well as maximizing the stability of the robot) does not pose a unique solution.
The red circles shown in Fig. 12 represents the pareto-optimal solutions of the multi-objective optimization problem. The blue curve joining those solutions is called as Pareto-Optimal front. Table 3 shows the solutions obtained by assigning different weights to the objective functions. From the table, it can be inferred that, when the weight of the energy function is tending towards 1, there is a greater optimization of energy achieved. For example, consider the fourth solution in Table. The energy function is assigned a weight of 0.8, as compared to the stability function, which has been assigned a weight of only 0.2. As a result, there is much greater amount of energy optimization achieved (approx. 67%). On the other hand, if the weight of the stability function is tending more towards 1 (first solution shown in Table), the stability of the robot increases by 78.22%, as compared to the energy function (assigned weight=0), which is optimized by only 7.66% of its default value. Thus, the main question here is, how does the user

Table 4 Joint motor current values of NaO robot

S/N	t_s	z_{cm}	A	B	$I_{total}(mA)$	% ↓ in energy
1	0.41	0.25	0.135	0.04	36.208	0 (default)
2	0.401	0.3216	0.1398	0.0252	34.21	5.52
3	0.4015	0.3215	0.11	0.021	23.04	36.36
4	0.417	0.3168	0.1082	0.0165	22.15	38.83
5	0.5938	0.1687	0.1045	0.0144	17.296	52.23

choose the weights of the respective functions. The answer to this is: High energy consumption by the robot is a serious problem, but the problem of stability overshadows it. This is because, if the robot is not stable, it cannot walk stably. If the robot fails to walk stably, there is no use of having a lesser energy consumption by the robot. Hence, the problem of stability of the robot is prime to the user as far as the problem of energy consumption is concerned.

That is why, it was decided to assign greater weight to the Stability function of the robot. It can be clearly seen from Table 3 that, when the Stability function is assigned a weight of 0.6 (solution 3 of the Table), both the objectives are sufficiently met. The stability value (ζ) is very close to 0, which indicates that the desired ZMP position (ZMP_{des}) almost overlaps the calculated ZMP position (ZMP_{calc}). Thus, assigning a greater weight to the stability function, greatly stabilizes the bipedal robot locomotion. On the other hand, as far as the the energy consumption is concerned, applying the same walk parameters to the energy function optimizes it by 46.74% of its default value. Now, based in the above discussion, one may look at the first solution of the Table 3 and argue that: How does the stability of the robot decreases as compared with the third solution (from 84.67 to 78.22%), even though the weight assigned to the stability function is more towards 1? The above question can be answered by carefully looking at the graph shown in the Fig. 12. A serious interpretation of it may reveal that, as the weight of the Stability function is increased,it means that the user wants the stability of the robot should be more and more maximized. However, the main equation defining stability does not force us to produce only positive values. It only states that, the distance between the ZMP trajectories should be minimized, in order to ascertain maximum stability. Thus, when the weight assigned to the stability function is more towards 1, it corresponds to the higher values of Stability axis in the Pareto front (shown in the Fig. 12). Any value of ζ deviating from 0 indicates that the robot is becoming unstable. For example, if the stability function is assigned a weight 0.9, the Stability function has a value close to -2.7×10^{-3}. This means that, the ZMP trajectories (ZMP_{calc} and ZMP_{des}) are far apart by a magnitude of 2.7×10^{-3}. As a result, the robot is less stable.

6 Results: application on a real NaO

Table 3 shows the theoretical value of Energy and Stability obtained by substituting the optimized value of the walk parameters. However, the energy consumed by the real robot varies. This is because, one of the major factors affecting the stability as well as energy consumed, is the impact that the robot's foot makes with the ground. Though the impact is not considered here, but still the result remains comparable with that obtained in theory.

To illustrate the reduction of energy consumed by NaO robot, it was made to walk a distance of 2 meters with the walk parameters specified in Table 3. In order to measure the energy consumed by the robot, the current values[3] of 12 different joints are taken. The joints are selected based on their usage in walking. For example, there are two joints located in NaO's head. However, the head joints do not play any role in its walking. Thus, the head joint current was ignored. The 12 joint motors, whose values are taken are: Left and Right Hip Roll motors, Left and Right Hip YawPitch motors, Left and Right Hip Pitch motors, Left and Right Knee pitch motors, Left and Right Ankle Roll motors and Left and Right Ankle Pitch motors. The table below shows the total current consumed by the robot for each set of optimized walk parameters shown in the the Table 3. The first row indicates the walk parameters and the total current consumed by 12 joint motors of the default Aldebaran walk.

It can be noted from Tables 3 and 4 that, the proposed theoretical reduction in energy varies, as compared to practical readings obtained from NaO robot. This is because, certain factors like energy loss occurred during impact of foot with the ground, change in armature resistance of motors due to heating etc. is not taken into consideration in the theoretical derivation. However, the practical results obtained for the reduction in energy consumption from real NaO robot, is comparable to that obtained in Table 3.

[3] obtain the motor current readings of NaO robot, the joint current sensor keys were selected in Choregraphe simulation software. When the robot walks, the current readings of the corresponding joints are stored in a *.csv file.

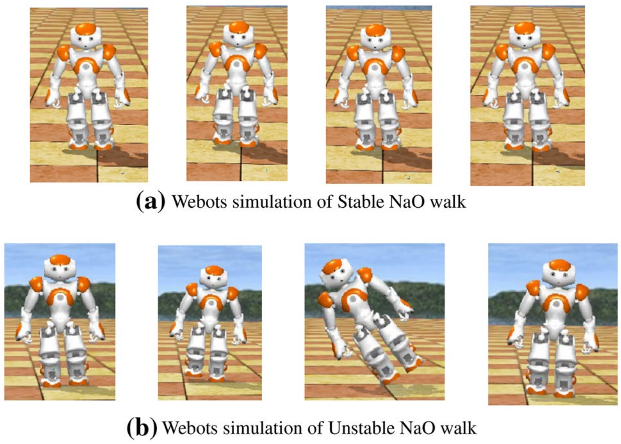

(a) Webots simulation of Stable NaO walk
(b) Webots simulation of Unstable NaO walk

Fig. 13 Simulation of NaO in Webots

To demonstrate the stability improvement in real NaO, the robot was made to walk with the optimized set of walk parameters as mentioned in Table 3. It was seen that, when NaO was made to walk with optimized parameters, specified in the fifth row of Table 3, there were large amount of lateral deviations in the robot, which ultimately led the robot to fall. This is shown in the Fig. 13. The experiments were conducted on a real NaO robot as well as simulations were carried out in Webots simulation software. Figure 13b demonstrates, NaO robot walking with the walk parameters mentioned in row 5 of Table 3. It is seen that, with these set of walk parameters, the amplitude of the lateral oscillations is large, which ultimately causes the robot to fall. In order to improve stability, the robot was made to move with the walk parameters mentioned in row 3 of Table 3. It is observed that, when NaO is made to walk with those walk parameters, the amplitude of the lateral deviation is very small.

7 Conclusion

We presented an approach to optimize walk parameters with multiobjective optimization for the better stability and minimum energy consumption for bipedal locomotion.The strong theoretical analytical method provides the necessary scalability and generic solution for the multiobjective problem for bipedal locomotion stability and energy consumption.Since generating the multiobjective function using the orbital energy and ZMP concepts was critical and this is the major contribution of this paper. Another contribution includes developing the single objective optimization function for stability and energy consumption and optimized by evolutionary genetic algorithm. Comparison of the results with policy gradient reinforcement learning and stiffness based walking shows that our method provides better stability and minimum energy consumption for biped locomotion. Future research may be directed towards increasing the walk parameters and consider the effect of the leg strike to the ground.

8 Future work

Future research may be directed towards increasing the walk parameters and consider the effect of the leg strike to the ground. Though, the stability function formulated, yields excellent improvement over the stability of the bipedal robot, but it has a disadvantage that, it does not take the step length into consideration. This is a problem, because when the step length (B) of a real NaO is increased to 0.09 m, the robot falls. Therefore, future work aims to build a stability function, that integrates the step length factor as well as considers the effect of impact, that occurs during leg-strike.The variation of the CoM also can be considered in the future to generalize the energy and stability functions.

References

1. Westervelt ER, Grizzle JJW, Chevallereau C, Choi JH, Morris B (2007) Feedback control of dynamic bipedal robot locomotion. CRC press, Boca Raton
2. Vukobratovic Miomir, Borovac Branislav (2004) Zero-moment pointthirty five years of its life. Int J Humanoid Robotics 1(01):157–173
3. Kuo Arthur D (2007) The six determinants of gait and the inverted pendulum analogy: a dynamic walking perspective. Human Mov Sci 26(4):617–656
4. Gong D, Yan J, Zuo G. A review of gait optimization based on evolutionary computation. Appl Comput Intell Soft Comput 2010:1–12, Article ID 413179
5. Goldberg David E, Holland John H (1988) Genetic algorithms and machine learning. Mach Learn 3(2):95–99
6. Capi G et al (2002) Optimal trajectory generation for a prismatic joint biped robot using genetic algorithms. Robot Auton Syst 38(2):119–128
7. Capi G et al (2003) Real time gait generation for autonomous humanoid robots: a case study for walking. Robot Auton Syst 42(2):107–116
8. Park JH, Choi M (2004) Generation of an optimal gait trajectory for biped robots using a genetic algorithm. JSME Int J Ser C Mech Syst Mach Elem Manuf 47(2):715–721
9. Choi SH, Choi YH, Kim JG (1999) Optimal walking trajectory generation for a biped robot using genetic algorithm. Intelligent Robots and Systems, 1999. IROS'99. In: Proceedings of 1999 IEEE/RSJ international conference on. Vol. 3. IEEE
10. Kulk JA, J S Welsh (2008) A low power walk for the NAO robot. In: Kim J, Mahony R (eds) Australasian conference of robotics and automation (ACRA), Canberra, 3–5 Dec 2008

11. Sun Z, Roos N (2013) An energy efficient gait for a Nao robot. In: BNAIC 2013: Proceedings of the 25th Benelux conference on artificial intelligence, Delft, The Netherlands, November 7–8, 2013. Delft University of Technology (TU Delft); under the auspices of the Benelux Association for artificial intelligence (BNVKI) and the Dutch research school for information and knowledge systems (SIKS)
12. Ames AD, Cousineau EA, Powell MJ (2012) Dynamically stable bipedal robotic walking with NAO via human-inspired hybrid zero dynamics. In: Proceedings of the 15th ACM international conference on Hybrid Systems: Computation and Control. ACM
13. Lin CM, Peng YF (2004) Adaptive CMAC-based supervisory control for uncertain nonlinear systems. Syst Man Cybern Part B Cybern IEEE Trans 34(2):1248–1260
14. Miller WT (1994) Real-time neural network control of a biped walking robot. Control Syst IEEE 14(1):41–48
15. Zhou Changjiu, Meng Qingchun (2003) Dynamic balance of a biped robot using fuzzy reinforcement learning agents. Fuzzy Sets Syst 134(1):169–187
16. Jha RK, Singh B, Pratihar DK (2005) On-line stable gait generation of a two-legged robot using a geneticfuzzy system. Robot Auton Syst 53(1):15–35
17. Udai AD (2008) Optimum hip trajectory generation of a biped robot during single support phase using genetic algorithm. In: Emerging trends in engineering and technology, 2008. ICETET'08. First international conference on, IEEE pp 739–744
18. Vundavilli PR, Sahu SK, Pratihar DK (2007) Online dynamically balanced ascending and descending gait generations of a biped robot using soft computing. Int J Humanoid Robot 4:777–814
19. Rajendra R, Pratihar DK (2015) Analysis of double support phase of biped robot and multi-objective optimization using genetic algorithm and particle swarm optimization algorithm. Sadhana 2:549–575 (no. 2)
20. Lee JY, Lee JJ (2004)Optimal walking trajectory generation for a biped robot using multi-objective evolutionary algorithm. In: Control conference, 2004. 5th Asian, IEEEvol. 1, pp 357–364
21. Dip G, Prahlad V, Kien PD (2009) Genetic algorithm-based optimal bipedal walking gait synthesis considering tradeoff between stability margin and speed. Robotica 27(03):355–365
22. Rajendra R, Pratihar DK (2012) Particle swarm optimization algorithm vs. genetic algorithm to solve multi-objective optimization problem in gait planning of biped robot. In: Proceedings of the international conference on information systems design and intelligent applications 2012 (INDIA 2012) held in Visakhapatnam, India, January 2012. Springer Berlin Heidelberg, pp 563–570
23. Capi G, Kaneko SI, Mitobe K, Barolli L, Nasu Y (2002) Optimal trajectory generation for a prismatic joint biped robot using genetic algorithms. Robot Auton Syst 38(2):119–128
24. Kajita Shuuji, Yamaura Tomio, Kobayashi Akira (1992) Dynamic walking control of a biped robot along a potential energy conserving orbit. Robot Autom IEEE Trans 8(4):431–438
25. Kajita S, Kanehiro F, Kaneko K, Yokoi K, Hirukawa H (2001) The 3D linear inverted pendulum mode: a simple modeling for a biped walking pattern generation. In: Intelligent robots and systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on, vol. 1. IEEE, pp 239–246
26. Lee JY, Kim MS, Lee JJ (2004) Multi-objective walking trajectories generation for a biped robot. In: Intelligent robots and systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ international conference on, vol. 4. IEEE, pp 3853–3858
27. Kumar S, Raj M (2015) Energy optimized trajectory generation for bipedal locomotion, in: International Conference on Computing in Mechanical Engineering (ICCME-2015) (ICCME-2015), Kochi, India, India
28. Deb K, Pratap A, Agarwal S, Meyarivan TA (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans 6:182–197 (no. 2)
29. Taormina R et al (2015) Data-driven input variable selection for rainfall-runoff modeling using binary-coded particle swarm optimization and extreme learning machines. J Hydrol 529(3):1617–1632
30. Zhang J et al (2009) Multilayer ensemble pruning via novel multi-sub-swarm particle swarm optimization. J Univ Comput Sci 15(4):840–858
31. Wang WC et al (2015) Improving forecasting accuracy of annual runoff time series using ARIMA based on EEMD decomposition. Water Resour Manag 29(8):2655–2675
32. Zhang SW et al (2009) Dimension reduction using semi-supervised locally linear embedding for plant leaf classication. Lect Notes Comput Sci 5754:948–955
33. Wu CL et al (2009) Methods to improve neural network performance in daily ows prediction. J Hydrol 372(1–4):80–93
34. Chau KW et al (2010) A hybrid model coupled with singular spectrum analysis for daily rainfall prediction. J Hydroinf 12(4):458–473
35. Choi Y, You BJ, Oh SR (2004) On the stability of indirect ZMP controller for biped robot systems. In: Intelligent robots and systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ international conference on, vol. 2. IEEE, pp 1966–1971
36. Choi Youngjin, Kim Doik, Yonghwan Oh, You Bum-Jae (2007) Posture/walking control for humanoid robot based on kinematic resolution of com jacobian with embedded motion. Robot IEEE Trans 23(6):1285–1293
37. Erbatur Kemalettin, Kurt Okan (2009) Natural ZMP trajectories for biped robot reference generation. Ind Electr IEEE Trans 56(3):835–845
38. Gouaillier D, Collette C, Kilner C (2010) Omni-directional closed-loop walk for NAO. In: Humanoid robots (Humanoids), 2010 10th IEEE-RAS international conference on. IEEE, pp 448–454