International Journal of Technical Research and Applications e-ISSN: 2320-8163,
www.ijtra.com, Special Issue 42 (AMBALIKA) (March 2017), PP. 105-112

ENERGY AND STABILITY ANALYSIS OF BIPED LOCOMOTION

[1] **Sachin Kumar Sonker, [2] Dr. Kamlesh Singh, [3] Dr. Alok Mishra , [4]Jai Pratap Dixit**
[1] M. Tech Scholar, [2] Associate Professor, [3] Director
[1,] Department Of Automation & Robotics, Ambalika Institute of Management & Technology, Lucknow
[2] Department of Mechanical Engineering, Ambalika Institute of Management & Technology, Lucknow.
[3] Department Of Applied Science, Ambalika Institute of Management & Technology, Lucknow.
[1] sachinsonker@hotmail.com, [2] mailkamleshsingh@gmail.com, [3] addl.director@aimt.edu.in, [4]jpdixit.iiita@gmail.com

Abstract— **This paper presents energy and stability functions, integrating the stability parameters of the Zero Moment Point (ZMP) classed bipeds where stability parameters are the positions of Center of Mass (CoM) and ZMP respectively. The Energy function is derived using the concept of Orbital Energy and is optimized using Real Coded Genetic Algorithm to produce an optimum set of walk parameters, which consumes minimum energy, during walking. A Stability function is also proposed, which is obtained by modifying the pre-existing ZMP trajectory. The ZMP trajectory is modified in such a manner, that it remains at the center of the convex hull, not only during the single support phase, but also during the transition of the robot from the Double Support Phase (DSP) to Single Support Phase (SSP) and vice-versa. 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.**

Index Terms— **Zero Moment Point, Center of Mass, Multi-Objective Optimization, Bipedal Locomotion, Pareto-optimal front.**

I. INTRODUCTION

Bipedal Locomotion is still an open ended problem and poses great challenges to the researchersfrom 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 a significant contribution 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 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 consumptionas well as increasing its stability. This section discusses the research carried out in the fieldof 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 optimizationis 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 off-line walking pattern generation. So, in order to design an energy efficient gait, the optimization is carried out before the actual walking starts.

II. LITERATURE SURVEY

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 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-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 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 thejoints. 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. EmploysPolicy 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 method 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, which 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 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.

The paper uses the concept of orbital energy to develop an energy function which indicates the energy consumed by the robot in one gait cycle. On the other hand, recent works on a similar background [19] used the concept of minimum change of torque at the joints to formulate an energy function. This paper also 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 [15],[18],[14], have attempted to build a stability function, which had only taken the stability concern of the robot during the SSP.

III. METHODS

A. Orbital Energy - Concept and Derivation

The Orbital Energy can be calculated only for those bipeds whose CoM follows the Potential Energy Conserving Orbit (PECO). PECO is defined as a particular class of trajectories of a biped robot in which the center of gravity of the body moves in a horizontal direction. The horizontal dynamics of the center of gravity can then be expressed by a simple linear differential equation. The important question is that, why the term Potential Energy Conserving Orbit and not Kinetic Energy or simply energy conserving orbit? This is simply, because, the constraint is imposed on the CoM position. Since, the CoM always needs to follow a straight horizontal line, located at a fixed height above the ground; therefore the potential energy always remains conserved. The Orbital Energy concept is energy based approach and this method is computationally very efficient. This is because, as seen in Section 4.2, the user specifies only two things. First, the energy required to execute support exchange between the legs i.e. the energy required to switch from DSP to SSP and vice-versa and second the length of a step. Based on these two parameters, the CoM position for the next support exchange phase is determined from equation. Thus, a particular sequences of orbital energies (E0, E1, E2…) and steps lengths (s1, s2, s3 …) correspond to a specific walk pattern

1) Expression for Orbital Energy

The relation of orbital energy with the CoM can be found by multiplying xcm on both sides of horizontal dynamics of CoM equation.

$$\ddot{x}_{cm}=\frac{g}{z_{cm}}x_{cm} \qquad \text{Equation 1}$$

Integrating the above equation 1it we get.

$$\int(\ddot{x}_{cm}\dot{x}_{cm}-\frac{g}{z_{cm}}x_{cm}\dot{x}_{cm})dt=constant=E$$

$$E=\frac{1}{2}\dot{x}_{cm}^2-\frac{g}{2z_{cm}}x_{cm}^2$$

Equation 2

In the above derivation, we have assumed energies per unit mass and therefore, the mass of CoMis not mentioned in equation 2. The first term refers to the kinetic energy per unit mass and the second term is imaginary. Equation 2 specifies that the orbital energy is conserved in the motion of Linear Inverted Pendulum Momentum (LIPM).
Since bipedal walking takes place in 3-D, equation 2 can be extended to 3-D by Kajita [26], which is given by-

$$E_x=-\frac{g}{2z_{cm}}(x_{cm}+y_{cm})^2+\frac{1}{2}(\dot{x}_{cm}+\dot{y}_{cm})^2 \qquad \text{Equation 3}$$

$$E_y=-\frac{g}{2z_{cm}}(-x_{cm}+y_{cm})^2+\frac{1}{2}(-\dot{x}_{cm}+\dot{y}_{cm})^2 \qquad \text{Equation 4}$$

Where Ex and Ey are the horizontal energy components in the forward and the lateral directions respectively.

2) Phase Portrait of Energy Trajectories

The phase portrait of orbital energy can be categorized under three different cases. The phase portrait of the energy trajectories is shown in the Figure 1.

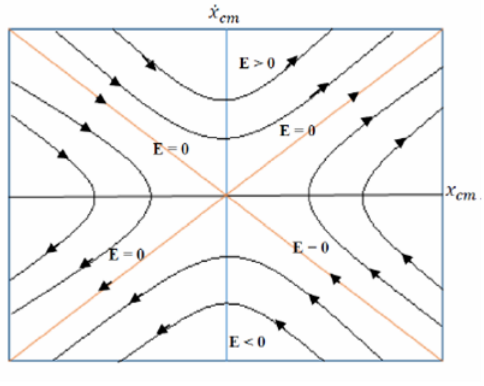


Figure 1:Phase portrait of Orbital Energy trajectories

Case 1: E < 0: When the orbital energy of the robot is less than zero, it never crosses the line xcm=0 (thepoint when the CoM of the robot is just above the supporting leg). This case is similar to the concept shown in Figure 1. If E < 0, the LIPM does not possess sufficient energy to cross a point located above the pivoting point at the ground. As a result, the CoM changes its moving direction, resulting in unsuccessful support exchange.

Case 2: E > 0: When the orbital energy of the robot is greater than zero, the body swings from negative to the positive side of xcm axis and vice-versa. This is similar to the concept explained in Figure 1. If E > 0, the LIPM has sufficient velocity, such that the CoM continues to move in the same direction. This situation corresponds to successful leg exchange.

Case 3: E = 0: When the orbital energy of the robot is equal to zero, it represents an equilibrium state. During this state, CoM of the body will be directly above the support foot. This is similar to the condition in LIPM, when the CoM is directly above the pivot. At this moment, xcm = 0 and xcm= 0.
Substituting E = 0 in equation 2 generates two eigen vectors

$$\dot{x}_{cm}=\pm x_{cm}\sqrt{\frac{g}{z_{cm}}} \qquad \text{Equation 5}$$

Equation 5 represents a saddle point with two eigen vectors, one stable and other unstable.When the distance between the CoM of the body and the CoP (located within the convex hull of the supporting foot) is decreasing, xcm and ˙xcm have opposite signs. This corresponds to stable eigen vector. On the other hand, when the distance between the CoM and CoP is increasing, xcmand ˙xcm have same sign. This corresponds to an unstable eigen vector.

B. Bipedal Walk Pattern Generation

This section discusses the methods to generate walking patterns of a bipedal robot in a realworld. For a ZMP based LIPM bipedal robots, the followingfacts holds true:
1. The robot walks with bent knees in order to prevent singularity problem.
2. During bipedal walk, the CoM of the robot maintains a constant height above the groundsurface.
3. The prismatic force in the LIPM does not demand physically the presence of a prismaticjoint. Instead, it means that the leg joints (knee, hip and ankle) be rotated in such a mannerthat, the rotational motion of these joint causes an effectivelinear translation of the CoM.
A walk pattern can be easily generated by splitting the 3-D LIPM trajectory into small parts.

1) Walk Primitive

Consider a part of the 3-D LIPM trajectory as shown in Figure 2, when the robot takes a step forward. The trajectory is defined for a period of t = 0 to t = ts, where ts is defined as the support exchange time. It can be inferred that the trajectory is hyperbolic in nature and symmetric about the y-axis. This part of the trajectory is called as Walk Primitive. The walk primitive of a bipedal robot can be found out easily, if the support exchange time ts and the height of the plane to which the torso is constrained is given, i.e. zcm. Given these two parameters, the terminal position as well as the terminal velocity of a walk primitive i.e. (xs,ys) and (vxs,vys) can be calculated. The equation representing the walk primitive can be derived from equation 11 and is given by:

2) Walk Parameters

In order to generate walk patterns for a biped on a known ground profile, the walk parameters are set before, the robot actually starts walking. There are basically four walk parameters of a ZMP classed bipedal robot. They are:

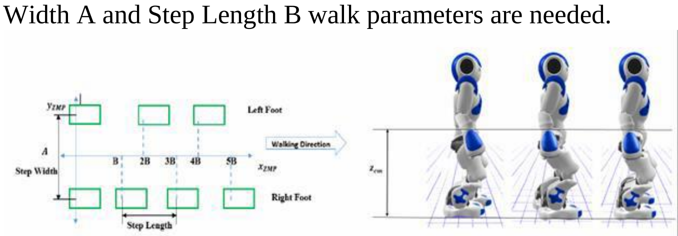
1. Step Length (B): As the name indicates, this refers to the length of the step taken by the bipedal robot, while walking in the forward direction. This is illustrated in the Figure 2(a).

2. Step Width (A): This refers to the lateral distance present between the feet of the bipedal robots as shown in the Figure 2(a).

3.CoM Height (height of the constraint plane(zcm): This represents the height of the plane to which the CoM of the robot (situated near Torso part) is constrained throughout the walking cycle as shown in Figure 2(b).

4. Support Leg Exchange Time (ts): During walking, the robot goes a transition from DSP to SSP or from the left to right support leg and vice-versa. Here ts refer to the time required to either exchange the support legs or change from DSP to SSP (vice-versa).

In order to plan appropriate foot placement off-line only Step Width A and Step Length B walk parameters are needed.



(a) NAO Robot Footprints illustrating walk parameters (b) Constant CoM Height of NAO

Figure 2: Walk Parameters of a Bipedal Robot

3) Algorithms for generating 3-D Walking Pattern

This section explains the step-wise procedure for generating bipedal walking pattern based on LIPM concept.

Step 1: Initially set the walk parameters sx, sy of the robot and the support foot time period ts. Set the position of the CoM (x; y) and the initial foot position (Xs(0),Ys(0)).

Step 2: t = 0, n = 0

Step 3: Integrate the equation of LIPM given by equation 10 as well as along the y-axis (by replacing all the x-component terms in equation 10 by its equivalent y-component) from t to t+ts.

Step 4: t = t + ts, n = n + 1

Step 5: Calculate the foot position for the next step, by using equation 21.

Step 6: Compute the walk primitive for the next step , by using equation 22 and 23.

Step 7: Calculate the target state (xt; ˙xt) from the corresponding equation:

$$\begin{bmatrix} x^t \\ \dot{x}^t \end{bmatrix} = \begin{bmatrix} X_s^{(n)} + x_s^{(n)} \\ v_{ss}^{(n)} \end{bmatrix} \quad \text{Equation 6}$$

In a similar manner, calculate the target state (yt; ˙yt).

Step 8: Check if target position is reached. If not, go back to Step 3.

C. Stability Parameters of Bipedal Robots: 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 [27] 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.

D. Formulation of the Objective Function

The stability and energy functions are mainly derived by modifying the ZMP and CoM trajectories [29],[30], [31]. This section discusses the derivation of the Energy function using the concept of Orbital Energy. Also, the Stability function is derived by slightly modifying the ZMP trajectory. Since, there is more than one objective function, therefore the optimization of the two different functions, results in two different set of optimized parameters. However, the user needs the robot to be energy efficient as well as have maximum stability both at the same time.

1) The Energy Function

Mostly, the energy functions derived for a bipedal robot uses the concept of the change in torque at the joints [29],[6],[30],[4]. 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 be-cause, 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).

The orbital energy of the bipedal robot changes from one support phase to another [32]. This is quite intuitive, since the generalized expression for energy given by equations 14 and

15 shows its dependence on ZMP and CoM, which in turn varies during the transition of the support foot. Therefore, we use equations 14 and 15 and substitute the corresponding values of CoM and ZMP to obtain the value of energy at di_erent time instants. A generalized version of the equation 14 and 15 can be re-written as:

$E_{x(i)}(t) = -\frac{g}{2z_{cm}}(x_{cm(i)} + y_{cm(i)})^2 + \frac{1}{2}(\dot{x_{cm}} + \dot{y_{cm}})^2$ Equation 7

$E_{y(i)}(t) = -\frac{g}{2z_{cm}}(x_{cm(i)} + y_{cm(i)})^2 + \frac{1}{2}(-\dot{x_{cm}} + \dot{y_{cm}})^2$ Equation 8

Where

$$i = \begin{cases} 1. & if \quad 0 \le t \le ts \\ 2. & if \quad ts < t < T - ts \\ 3. & if \quad T - ts \le t \le T \end{cases}$$

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

Case 1: $0 \leqslant t \leqslant ts$

During this time interval, the robot is in DSP. The CoM and ZMP trajectories are almost similar which is given by equation 28 and 32. The equation for orbital energy can therefore be obtained by substituting the values of xcm and ycm in the equation 7 and 8 as:

$x_{cm}\ \frac{Kx}{td}t$; $y_{cm}\ \frac{Ky}{td}t$ Equation 10

Substituting xcm(1) = xcm and ycm(1) = ycm from equation and x'cm(1) $\frac{\partial}{\partial t}$xcm(1) and y'cm(1) $\frac{\partial}{\partial t}$ycm(1) in equation 7 and 8, Ex1 and Ey1 can be calculated. E1 i.e. the energy during this interval, is then computed as:

$E1(t) = \sqrt{(Ex1(t))^2 + (Ey1(t))^2}$ Equation 11

$E_1 = \int_{t=0}^{t=ts} E1(t)dt$ Equation 12

Case 2: ts ≤ t ≤ T - ts

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

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

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 equation 7 and 8.

$x_{cm}(t) = \frac{B}{T}(t - \frac{T}{2}) + \sum_{n=1}^{n=\infty} \frac{BT2\omega2(1+\cos n\pi)}{n\pi(T2\omega2+n2\pi2)} \sin\frac{n\pi}{T}t$ Equation 13

$y_{cm}(t) = \sum_{n=1}^{n=\infty} \frac{2AT2\omega2(1-\cos n\pi)}{n\pi(T2\omega2+n2\pi2)} \sin\frac{n\pi}{T}t$ Equation 14

Substituting xcm(2) = xcm and ycm(2) = ycm from equation and x'cm(2) = $\frac{\partial}{\partial t}$xcm(2) and y'cm(2) = $\frac{\partial}{\partial t}$ycm(2) in equation 12 and 13, Ex2 and Ey2 can be calculated. E2 i.e. the energy during this interval, is then computed as:

$E_2(t) = \sqrt{(Ex2(t))^2 + (Ey2(t))^2}$ Equation 15

$E_2 = \int_{t=ts}^{t=T-ts} E2(t)dt$ Equation 16

Case 3: T - ts ≤ t ≤ 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 Figure 3, and is given by:

$x_{cm}(t) = (2B - k_x) + \frac{kx}{ts}(t - (T - t_s)) \triangleq x_{zmp}(t)$ Equation 16

$y_{cm}(t) = (\frac{ky}{ts})(T - t) \triangleq y_{zmp}(t)$ Equation 17

Substituting xcm(3) = xcm and ycm(3) = ycm from equation and x'cm(3) $\frac{\partial}{\partial t}$xcm(3) and y'cm(3) $\frac{\partial}{\partial t}$ycm(3) in equation 16 and 17, Ex3 and Ey3 can be calculated. E3 i.e. the energy during this interval, is then computed as

$E_3(t) = \sqrt{(Ex3(t))^2 + (Ey3(t))^2}$ Equation 18

$E_3 = \int_{t=T-ts}^{t=T} E3(t)dt$ Equation 19

From the above discussions, it can be concluded that energy is a function of x_{zmp}, y_{zmp}, z_{cm} and ts. 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:

$E(t_s, z_{cm}, A, B) = \frac{1}{3}\sum_{j=1}^{j=3} Ej$ Equation 20

The aim is to minimize the energy consumed, given by equation 20, subjected to the constraints specified by Aldebaran Robotics (manufacturer of NAO robot):

$0.14m < z_{cm} < 0.33m$; $0.4sec. \le t_s \le 0.6sec.$
$0.101m \le A \le 0.160m$; $0.01m \le B \le 0.08m$

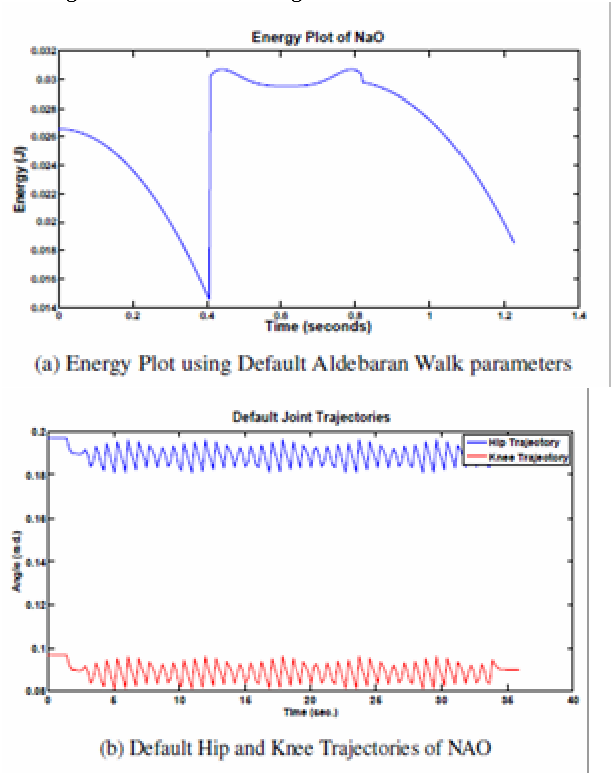
a) Optimization of the Energy Function - Results

RCGA is used to optimize the energy function given by equation 50, 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 plot obtained is also compared with the PGRL and sti_ness 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 [33] had the values of input parameters set as:

$t_s = 0.41sec$ $z_{cm} = 0{:}25m$ $A = 0.135m$ $B = 0.04m$

The energy plot of the default Aldebaran walk over a time period T is shown in the Figure 3a, the energy consumed by the robot during walking was 0.0261 Joules. Figure



(a) Energy Plot using Default Aldebaran Walk parameters

(b) Default Hip and Knee Trajectories of NAO

Figure 3: Energy Plot and Joint trajectories of Default walk

3a shows that the energy consumed by the robot initially is high. A discontinuity in the energy graph is seen at time t = ts = 0.41sec, 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 equations 5 and 6 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 - ts), 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. Fig. 3b shows the hip and the knee trajec-tories of a real NaO robot using default walk parameters.

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. 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). 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.

3) Derivation of the Stability Function

The Stability function ζ) is given by the difference of the y-component of calculated ZMP position ZMPcalc and the desired ZMPdes (explained in this section). Mathematically, this can be representedas:

$\zeta = \| ZMP_{calc} - ZMP_{des} \|$ Equation 21

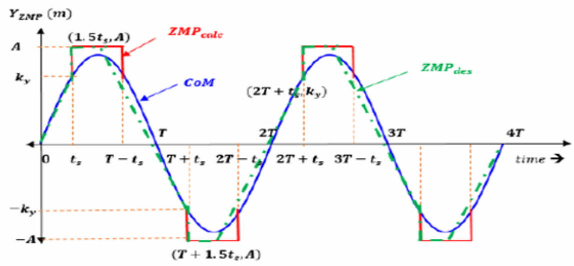


Figure 4: CoM, Calculated ZMP and Desired ZMP Trajectories of a Bipedal Robot

Lower values of 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 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 Figure 4 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 ZMPcalc is defined by equation 21, the only unknown in equation 21 is ZMPdes. the ZMP trajectory is modified and its equation is formulated in different time intervals.

Case 1: 0 ≤ t ≤ ts

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 + 1.5ts ≤ t ≤ 2T + ts. Thus,

$$y_{des}^{zmp}(t) = \frac{2}{5ts}(A + ky)t + c$$

At t = ts, y_{des}^{zmp}= ky. With these conditions, solving for constant c, the equations for desired ZMP trajectory can be written as:

$$y_{des}^{zmp}(t) = \frac{2}{5ts}(A + ky)t - \frac{2}{5}A + \frac{3}{5}ky \qquad \text{Equation 22}$$

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

$$y_{calc}^{zmp}(t) = \frac{ky}{ts}t \qquad \text{Equation 23}$$

Case 2: ts ≤ t ≤ 1.5ts

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:

$$y_{des}^{zmp}(t) = A \qquad \text{Equation 24}$$

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

$$y_{calc}^{zmp}(t) = A \qquad \text{Equation 25}$$

Case 3: 1.5ts ≤ t ≤ T + ts

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:

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

From the graph shown in Figure 4, at time t = 1.5ts => y_{des}^{zmp}(t) = A. Thus, solving for the unknown the desired ZMP equation during this interval is:

$$y_{des}^{zmp}(t) = \frac{2}{5ts}(A + ky)t + \frac{8}{5}A + \frac{3}{5}ky \qquad \text{Equation 26}$$

During this interval, the equation for y_{calc}^{zmp} is spitted in 2 time intervals i.e. 1.5ts ≤ t ≤ T - ts and T - ts ≤ t ≤ T : During the interval, 1.5ts ≤ t ≤ T - ts, the equation for y_{calc}^{zmp} is :

$$y_{calc}^{zmp}(t) = \frac{ky}{ts}t \qquad \text{Equation 27}$$

During the interval, T - ts ≤ t ≤ T, the equation for y_{calc}^{zmp} is :

$$y_{calc}^{zmp}(t) = \frac{ky}{ts}(T-t) \qquad \text{Equation 28}$$

Combining equations 22,24 and 26 into one single piece-wise equations, the equation for y_{des}^{zmp} can be obtained as:

$$y_{des}^{zmp}(t) = \begin{cases} \frac{2}{5t_s}(A + k_y)t - \frac{2}{5}A + \frac{3}{5}k_y, & if \quad 0 \le t \le t_s \\ A, & if \quad t_s < t < 1.5t_s \\ -\frac{2}{5t_s}(A + k_y)t + \frac{8}{5}A + \frac{3}{5}k_y, & if \quad 1.5t_s \le t \le T \end{cases} \qquad \text{Equation 29}$$

Once the equation for $y_{des}^{zmp}(t)$ is formulated, the Stability function can be obtained from the equation 21 as:

$$\zeta = \begin{cases} \frac{3k_y}{5t_s}t - \frac{2A}{5t_s}t + \frac{2}{5}A - \frac{3}{5}k_y, & if \quad 0 \le t \le t_s \\ 0, & if \quad t_s \le t \le 1.5t_s \\ \frac{2}{5t_s}(A + k_y)t - \frac{3}{5}A - \frac{3}{5}k_y, & if \quad 1.5t_s < t < T - t_s \\ -\frac{k_y}{t_s}T - \frac{3k_y}{5t_s}t + \frac{2A}{5t_s}t - \frac{8}{5}A - \frac{3}{5}k_y, & if \quad T - t_s \le t \le T \end{cases} \qquad \text{Equation 30}$$

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:

0.14m < zcm < 0.33m ; 0.4sec. ≤ ts ≤ 0.6sec.
0.101m ≤ A ≤ 0.160m ; 0.01m ≤ B ≤ 0.08m

The minimization of means that the ZMP trajectory remains at the center of the support poly-gon throughout the walking cycle. In order to realize the ZMP trajectory given by equation 29 on a real NAO robot, the CoM equation needs to computed by inputting the ZMPdes quations into equation 15.

a) Optimization of the Stability Function - Results

The Stability function given by equation 60 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 [33] had the values of the input parameters of NAO robot set as:

ts = 0.41sec zcm = 0.25m A = 0.135m B = 0.04m

The Stability plot of the default Aldebaran walk over one gait cycle is shown in Figure 5 . Using equation 60, 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 ZMPdes from the ZMPcalc trajectory. From Figure 5 , it can be inferred that, when the robot is switching.

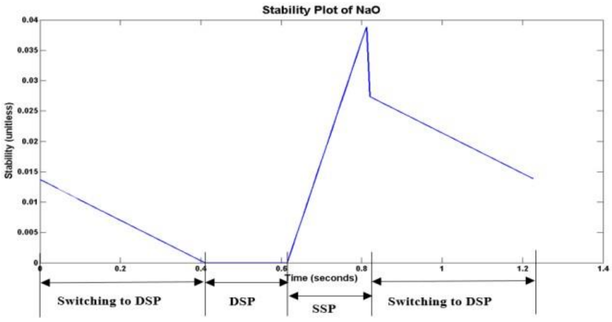


Figure 5: Stability Plot using Default Aldebaran Walk parameters

to DSP, the stability of the robot increases. This can be seen from the decreasing value of during the interval 0⩽ t ⩽ 0.41. The robot comes to complete DSP during the interval 0.41⩽ t ⩽ 0.6. During this time interval, the robot possesses maximum stability (= 0). In the next time interval 0.6⩽ t⩽_ 0.82, the robot switches to SSP. The SSP is a highly unstable state. This can

be seen from the sudden rising value of as shown in the Figure 5. During the time interval, 0.82⩽ t ⩽ 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 , as shown in the Figure 5. The stability function was then optimized using RCGA. The results obtained by using GA is

shown in Figure 6. 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:

ts = 0.6sec z_{cm} = 0.161m A = 0.101m B = 0.01m

The stability plot of the optimized walk is depicted from the bottom right figure 4. Substituting the above walk parameters in the equation 60, 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 Figure 4, 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 Figure 6.

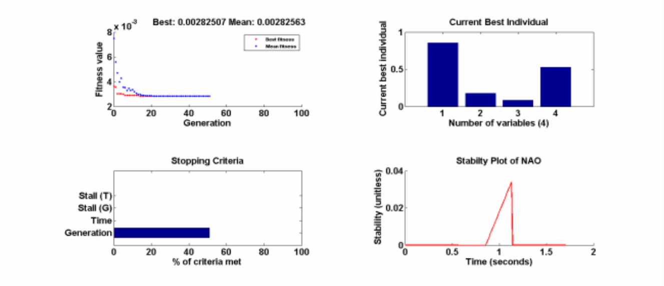


Figure 6: Optimization of Stability Function using RCGA

Conclusion

This paper presents an energy function, which is developed using the concept of orbital energy. The energy equation designed, is a function of four walk parameters. The energy function is then optimized using RCGA to yield an optimized set of walk parameters. The optimized walk parameters are then used to generate the joint trajectories of the robot, which consumes minimum energy. The energy optimization results obtained are compared with the results obtained through policy gradient reinforcement learning [11] and Stiffness based walking [10]. The theoretical results obtained, are superior to that obtained in [10] and [11]. Next, a stability function is developed, which aims to keep the ZMP trajectory at the center of the convex hull throughout the walking cycle. The stability function is then optimized using RCGA. It was found that, when the energy function is optimized, the stability of the robot decreases. At the same time, when the stability of the robot is increased, the energy consumed by the robot also increases. Thus, there is a clear trade-o_ between the two objectives. This is called Multi-Objective Optimization Problem. To solve this problem, a pareto-front is generated, which determines the best solution, such that both the objectives are satisfied to a certain level. An experiment is conducted on a real NAO robot, which justifies the theoretical calculations.

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 into consideration the step length into consideration. This is a problem, because when the step length (B) of a real NAO is increased to 0:09m, 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

References

[1] E. R. Westervelt, J. W. Grizzle, C. Chevallereau, J. H. Choi, B. Morris, Feedback control of dynamic bipedal robot locomotion, Vol. 28, CRC press, 2007.

[2] B. Borovac, Zero-moment point thirty five years of its life 1 (1) (2004) 157–173.

[3] A. D. Kuo, The six determinants of gait and the inverted pendulum analogy: A dynamic walking perspective, Human movement science 26 (4) (2007) 617–656.

[4] D. Gong, J. Yan, G. Zuo, A review of gait optimization based on evolutionary computation, Applied Computational Intelligence and Soft Computing 2010.

[5] D. E. Goldberg, J. H. Holland, Genetic algorithms and machine learning, Machine learning 3 (2) (1988) 95–99.

[6] G. Capi, S. Kaneko, K. Mitobe, L. Barolli, Y. Nasu, Optimal trajectory generation for a prismatic joint biped robot using genetic algorithms, Robotics and Autonomous Systems 38 (2) (2002) 119 – 128.doi:http://dx.doi.org/10.1016/S0921-890(01)00177-4. URL http://www.sciencedirect.com/science/article/pii/S0921889001001774

[7] G. Capi, Y. Nasu, L. Barolli, K. Mitobe, Real time gait generation for autonomous humanoid robots: A case study for walking, Robotics and Autonomous Systems 42 (2) (2003) 107 – 116. doi:http://dx.doi.org/10.1016/S0921-8890(02)00351-2.

URLhttp://www.sciencedirect.com/science/article/pii/S0921889002003512

[8] J. H. Park, M. Choi, Generation of an optimal gait trajectory for biped robots using a genetic algorithm, JSME International Journal-Series C-Mechanical Systems Machine Elements and Manufact 47 (2) (2004) 715–721.

[9] S.-H. Choi, Y.-H. Choi, J.-G. Kim, Optimal walking trajectory generation for a biped robot using genetic algorithm, in: Intelligent Robots and Systems, 1999. IROS '99. Proceedings. 1999 IEEE/RSJ International Conference on,Vol. 3, 1999, pp. 1456–1461 vol.3. doi:10.1109/IROS.1999.811684.

[10] J. Kulk, J. Welsh, A low power walk for the nao robot, in: In Proceedings of ACRA, Citeseer, 2008.

[11] Z. Sun, N. Roos, An energy e_cient dynamic gait for a nao robot, in: Autonomous Robot Systems and Competitions (ICARSC), 2014 IEEE International Conference on, 2014, pp. 267–272.

doi:10.1109/ICARSC.2014.6849797.

[12] A. D. Ames, E. A. Cousineau, M. J. Powell, Dynamically stable bipedal robotic walking with nao via humaninspired hybrid zero dynamics, in: Proceedings of the 15th ACM international conference on Hybrid Systems: Computation and Control, ACM, 2012, pp. 135–144.