Robotica (2018) volume 36, pp. 945–970. © Cambridge University Press 2018
doi:10.1017/S0263574718000188

Trajectory generation and step planning of a 12 DoF biped robot on uneven surface

Gaurav Gupta* and Ashish Dutta

Department of Mechanical Engineering, Indian Institute of Technology Kanpur, Kanpur 208016, Uttar Pradesh, India. E-mail: adutta@iitk.ac.in

(Accepted February 3, 2018. First published online: February 26, 2018)

SUMMARY

One of the primary goals of biped locomotion is to generate and execute joint trajectories on a corresponding step plan that takes the robot from a start point to a goal while avoiding obstacles and consuming as little energy as possible. Past researchers have studied trajectory generation and step planning independently, mainly because optimal generation of robot gait using dynamic formulation cannot be done in real time. Also, most step-planning studies are for flat terrain guided by search heuristics. In the proposed method, a framework for generating trajectories as well as an overall step plan for navigation of a 12 degrees of freedom biped on an uneven terrain with obstacles is presented. In order to accomplish this, a dynamic model of the robot is developed and a trajectory generation program is integrated with it using gait variables. The variables are determined using a genetic algorithm based optimization program with the objective of minimizing energy consumption subject to balance and kinematic constraints of the biped. A database of these variables for various terrain angles and walking motions is used to train two neural networks, one for real-time trajectory generation and another for energy estimation. To develop a global navigation strategy, a weighted A* search is used to generate the footstep plan with energy considerations in sight. The efficacy of the approach is exhibited through simulation-based results on a variety of terrains.

KEYWORDS: Biped robot; Genetic algorithm; Neural network; Trajectory generation; Step planning.

1. Introduction

One of the distinct advantages of legged systems is mobility, i.e., their ability of traverse diverse terrain. They can utilize isolated foothold locations to optimize support and traction.

Development of step plan at a higher level and a local trajectory generation module that can execute the desired sequence of steps while maintaining balance comprise two major challenges to be overcome while developing a locomotion strategy for a biped robot. At both the levels of planning, an *optimal* solution is desired, the criteria for establishing optimality vary with the application and the desired system behavior. In this work, these two seemingly independent problems are solved alongside each other to generate a locomotion scheme. The optimization for trajectory generation is performed using genetic algorithm (GA), while the step-planning problem is solved using an A* search. Literature addressing the problems of trajectory generation and step planning are discussed henceforth.

The concept of zero moment point (ZMP) has been widely utilized in generating trajectories for humanoids. ZMP is defined as that point on the ground at which the net moment of the inertial forces and the gravity forces has no component along the horizontal axes.[1] The validity of using ZMP for walk on inclined ground has also been shown in ref. [2]. For trajectory generation of a biped robot, two major approaches have featured in recent works. The first method uses precise dynamic knowledge of the system, such as the location of the center of masses and inertias of the links and motors for gait synthesis purpose. Huang *et al.*[3] used this approach to generate foot and hip trajectories in the sagittal plane to satisfy ZMP requirement. The other approach realizes on limited dynamic information but on feedback control. One of the most prominent studies belonging to this group is the 3-D linear inverted

* Corresponding author. E-mail: ggupta9777@gmail.com

pendulum method used by Kajita *et al.*[4] Trajectory generation for HUBO-3[5] utilizes simple linear and sinusoidal curves for trajectory generation of the foot and the pelvis in independent Cartesian coordinates, and an external control system is called upon to implement the gait online. In the trajectory planning of biped *Lucy*,[6] angular momentum equations were used to estimate the unactuated upper body dynamics of the biped, and leg link trajectories were generated using polynomial functions in order to keep the reference upper body trajectory close to its *natural* state. More recently, vertical movement of this point mass has been shown to be effective in achieving a diverse range of step length as well as better push recovery behavior.[7]

Many diverse techniques have been attempted to overcome the challenge of walking on uneven surfaces for biped robots. Takanishi *et al.*[8] introduced the idea of virtual surfaces to develop walking patterns over uneven surfaces; they presented a method of trunk compensation for generating dynamically feasible trajectories. Vundavilli and Pratihar[9] used inverse dynamics learned, neural network-based gait planner to negotiate sloping surfaces. Manchester *et al.*[10] used model predictive control with exponential stabilization to achieve walking on rugged surfaces for a compass gait walker. A multi-objective gait optimization method for a humanoid robot's walking on slopes was proposed in ref. [11], using the robot and slope model an inverted pendulum model was used to parametrize hip and ankle trajectories. A multi-objective optimization based on speed and energy considerations was carried out to obtain the walking pattern. In ref. [12], a motion pattern generator for slope walking in 3-D dynamics using preview control of ZMP is proposed. The future ZMP locations are selected with respect to known slope gradient, and the trajectory of the center of mass (CoM) of the robot is generated by using the preview controller to maintain the ZMP at the desired location. Zheng *et al.*[13] use a universal stability criterion that checks whether the resultant of the gravity wrench and the inertia wrench of a robot lies in the convex cone of the wrenches resulting from contacts between the robot and the environment and generates a feasible CoM motion to generate walking on slopes.

Over recent years, GA has been the popular choice for trajectory generation of biped robots. Arakawa and Fukuda[14] used hierarchical optimization to determine energy-efficient gait cycles. One was the GA layer that minimized the power consumption, and the other was the EP layerthat minimized the configuration of the interpolated points. Sarkar and Dutta[15] used GA for generating energy-efficient trajectories for biped with compliant links for different step lengths and slopes. Lim *et al.*[16] used GA for generating stair-climbing motion for 12 degrees of freedom (DoF) biped robot using human motion data. Cardenas-Macial *et al.*[17] used GA for generatic energy optimal walking motion with closed loop stability for a 3 DoF biped robot. A review of GA based approaches for biped trajectory generation is presented by Gong *et al.*[18] GA is principally based on Darwin's theory of natural selection. This makes it a natural choice for generating biped trajectories in various environments. Also, the dynamic equations for systems with high DoF are high order, highly coupled, non-linear and involve search in multi-dimensional and irregular spaces, thereby making the application of gradient-based optimization very difficult. It is due to these reasons that GA is used in this study as well as several other successful implementations elsewhere.

Due to the computational challenges of generating optimal gait online, machine-learning techniques have also gained popularity in trajectory generation of humanoid systems. In the work by Capi *et al.*,[19] results of optimization by GA on joint angles to generate energy-efficient trajectories were used to train an RBFNN (radial basis function neural network) for different step lengths and step times on level ground. Nagasue *et al.*[20] used the K-nearest neighbors method for walking trajectory generation on a slope. Learning theory was applied to improve the database with experience. Ferreira *et al.*[21] used support vector regression and neural-fuzzy network in trunk compensation for balance control of a biped. Optimal gait was generated for ditch crossing robots by Vundavilli and Pratihar[22] using neural networks and fuzzy logic based gait planners. Liu *et al.*[23] used SVM controller to perform learning on small datasets for gait control. Using foot and hip trajectories as inputs, trunk trajectories were predicted using SVM-based controller to generate balanced gaits.

To develop a footstep plan, Chestnutt *et al.*[24] formulated the problem in the form of a search tree using a discrete set of step transition. An A* search was then performed to develop a *globally optimum* navigation strategy. The cost of each step was expressed heuristically using the prospective *location* of the step. A tiered planning strategy was developed by Chestnutt and Kuffner,[25] at a higher level, graphs were created manually and a local search was performed to join two vertices on these graphs. More recently, Huang *et al.*[26] have incorporated *energetics* in step planning. The cost of a step, however, was not evaluated using dynamic calculations, rather polynomial functions were

https://doi.org/10.1017/S0263574718000188 Published online by Cambridge University Press

Table I. Various biped step planning methods.

Reference	Tilt ground	Stair climbing	Cost function	Dynamics considered	Planning method
24	Yes	Yes	Location based (heuristic)	No	A*
26	Yes	Yes	Energy + location (approx. + heuristic)	No	A*
27	No	No	Euclidean	No	ARA*/wA*
28	No	Yes	Step length based (guided by APF)	No	Best-First
34	No	No	–	Yes	RRT
30	No	Yes	Orientation + height	No	A*
33	No	No	Time minimization	No	VG
31	No	Yes	Euclidean + heuristic	No	Djikstra + A*
Current work	*Yes*	*No*	*Energy based*	*Yes*	*A*/wA**

utilized for curve fitting and energy estimation. A gradient cost was estimated heuristically along with other location-based costs. Hornung *et al.*[27] used Euclidean distance-based cost for each step and used graph searches such as A*, wA*, R* and ARA* for step planning. The motion primitives were dynamically expanded to enhance reachability to the goal position. In the work by Li *et al.*,[28] the biped step planning (BSP) was modeled as a two layered problem. At a global level, best-first planning was used for performing the graph search, and an artificial potential field was used as heuristic to guide the search. The path obtained from this greedy search was fed to a local level planner where foot trajectory was generated online using Bezier curves. Planning in a 2.5-D map was performed with the objective of step-minimization in a study by Cupec *et al.*[29] Here, a path was generated using depth and time limited A* search, and this path was then followed by a constrained step plan. The work by Gutmann *et al.*[30] also utilizes A*-based graph search in a 2-D environment. The cost of movements was set heuristically in terms of orientation changes, type of movement and terrain location. Candido *et al.*[31] also used a multi-layer planner for BSP. A global planner was used to decompose the workspace into simpler convex regions. A subgoal planner utilized Dijkstra hierarchical planner for connecting these regions using edges. Apart from Euclidean costs, transition cost was imposed on the movements from one region to another. Last, motion primitives were used to follow this path closely. Prasanth and Sudheer[32] discussed an approach for stepping over smaller obstacles using a predetermined step length and subsequent selection of a suitable trajectory to avoid the obstacle.

Soo-Hyun *et al.*[33] applied a roadmap-based approach using visibility graphs in a 2-D environment with the goal of minimizing the time, where paths involving a higher number of turns were penalized. Perrin *et al.*[34] used offline computed swept-volume approximations for quick collision checking during the planning process. The trajectories were generated using an inverted pendulum model for walking on level ground. RRT was used for planning the path with a branching factor of 276. A summary of some of the methods discussed thus far is represented in Table I.

While various groups have developed strategies for step planning, insufficient attention has been paid toward planning global navigation strategies for biped robotics with the goal of minimizing energy consumption. On the few occasions when energetics was considered, it was using overly simplistic approaches. Also, the role of robot dynamics in step planning has largely been ignored. Trajectory generation and step planning are two imperative aspects of biped locomotion, but work on these two has almost always been performed in isolation with one another. In this work, a study on biped dynamics and energetics for a variety of movements has been performed and then utilized for developing a global planning scheme.

A short overview of the approach is presented in Fig. 1. The main components of the approach are as follows:

- Kinematics and dynamics: Involves the development of the inverse kinematics and euler-lagrange formulation for robot manipulators. These are presented in Sections 2 and 3, respectively.
- Optimization using GA: The kinematic and dynamics models are used along with step parameters, such as slope (m), step-lengths (sl_1, sl_2), turning angle (α) and swing-foot(sf) to pose the optimization problem. This is solved using a GA-based approach in Section 5 and a database of such solutions is created.

https://doi.org/10.1017/S0263574718000188 Published online by Cambridge University Press

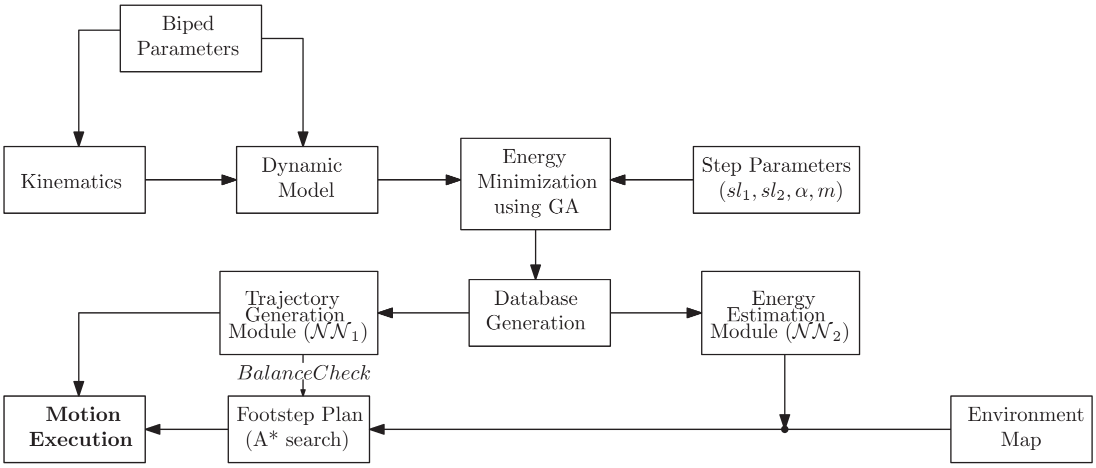

Biped Parameters
Kinematics
Dynamic Model
Energy Minimization using GA
Step Parameters (sl_1, sl_2, α, m)
Trajectory Generation Module ($\mathcal{NN}_1$)
Database Generation
Energy Estimation Module ($\mathcal{NN}_2$)
BalanceCheck
Motion Execution
Footstep Plan (A* search)
Environment Map

Fig. 1. Summary of the locomotion strategy.

- Trajectory generation module: A neural network ($\mathcal{NN}_1$) is trained to learn the mapping between the input parameters $(m, \alpha, sl_1, sl_2, sf)$ and the outputs (f_i) for real-time trajectory generation.
- Energy estimation: Another neural network ($\mathcal{NN}_2$) is trained using the same database and the same inputs to estimate the consumption of the step (E). This real-time energy estimation is used as a cost for the step-planner. Details on the two neural networks are available in Section 6.
- Step planner: A graph search is performed using a finite set of transitions represented by step parameters. These step parameters are a super-set of the parameters used for GA-based optimization. The search is performed using an A* search. As a final step, the step plan is executed using $\mathcal{NN}_1$. Graph search is described in Section 7.

2. Inverse Kinematics

The robot has a total of 12 DoF, 6 in each leg. The ankle joint can move in sagittal and frontal planes; knee joint has 1 DoF, and it can only move in the sagittal plane; and the hip joint has 3 DoF, and it allows movement in sagittal, frontal as well as transverse planes. The local frame is located on the ankle joint (P_0), with $\hat{x}_0$ pointing in the direction of walking, $\hat{z}_0$ pointing perpendicularly outward from the supporting surface and $\hat{y}_0$ is obtained by using the right-hand thumb rule. Since all the joints are revolute, $\hat{z}_i$ denotes the rotation axis for $i \in \{1 \ldots 12\}$ and $\hat{x}_i$ is normal to the plane of $\hat{z}_i$ and $\hat{z}_{i+1}$. Figure 3 illustrates the chosen coordinate frames with the help of a stick diagram. The DH parameters for all the transformations are listed in Table II; the convention of Craig[35] has been used. The joint angles are depicted in Fig. 2.

For performing the inverse kinematics, the positions and orientations of the reference points on the two feet (centroid of the supporting surface on the foot) and mid-hip serve as inputs. The rotation matrix needed to express the orientation of the local reference frame in terms of the global coordinate frame (${}^{global}_{ref}R$) is as follows:

$$ {}^{global}_{ref}R = R_{x,\phi} R_{y,\theta} R_{z,\psi} \tag{1}$$

Here ϕ, θ and ψ are euler angles. Using Eq. (1) and position X_{ref} of the reference point, the homogenous transformation matrix ${}^{global}_{ref}T$ is obtained. Subsequently, the transformation matrix at ankle joint can also be obtained.

Figure 3 illustrates the aforementioned coordinate frames clearly. In order to perform the inverse kinematics, the position of the hip joint X_h is expressed in terms of the base frame located at the ankle joint. (Using the position and orientation of the mid-hip, the task space coordinates of each hip joint can be obtained uniquely.)

$$ {}^{0}X_h = {}^{ref}_{0}T \; {}^{ref}X_h \tag{2}$$

https://doi.org/10.1017/S0263574718000188 Published online by Cambridge University Press

Table II. Link parameters for biped.

i	a_{i-1}	α_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	0	$-\pi/2$	0	θ_2
3	l_1	0	0	θ_3
4	l_2	0	0	θ_4
5	0	$\pi/2$	0	$\pi/2+\theta_5$
6	0	$\pi/2$	0	θ_6
7	l_3	0	0	θ_7
8	0	$-\pi/2$	0	$\pi/2+\theta_8$
9	0	$\pi/2$	0	θ_9
10	l_4	0	0	θ_{10}
11	l_5	0	0	θ_{11}
12	0	$-\pi/2$	0	θ_{12}

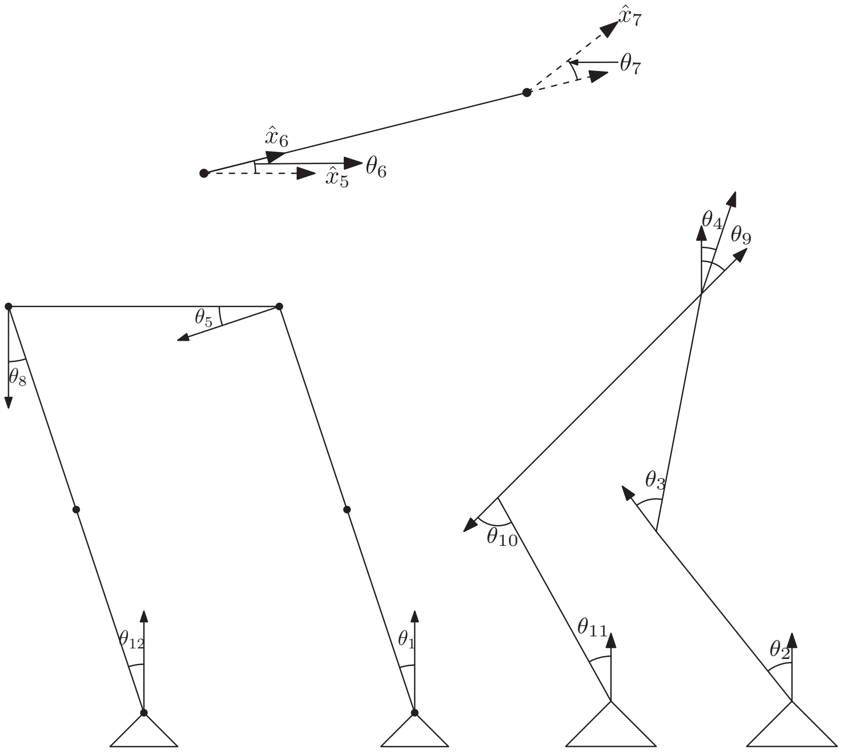


Fig. 2. Joint variable θ_i in frontal, sagittal and transverse planes.

The joint angle θ_1 is then given by

$$\theta_1 = atan2({}^{0}X_{h_x}, {}^{0}X_{h_y}) \tag{3}$$

Also,

$${}^{global}_{1}T = {}^{global}_{0}T\;{}^{0}_{1}T \tag{4}$$

Here, ${}^{0}X_{h_x}, {}^{0}X_{h_y}$ are the *x, y* components of ${}^{0}X_h$, respectively. In ΔABC as indicated in Fig. 4, $d = \sqrt{{}^{1}x_h^2 + {}^{1}z_h^2}$ and $\gamma = \cos^{-1}(\frac{l_1^2-d^2+l_2^2}{2l_1l_2})$. Therefore,

$$\theta_3 = \pi - \gamma \tag{5}$$

https://doi.org/10.1017/S0263574718000188 Published online by Cambridge University Press

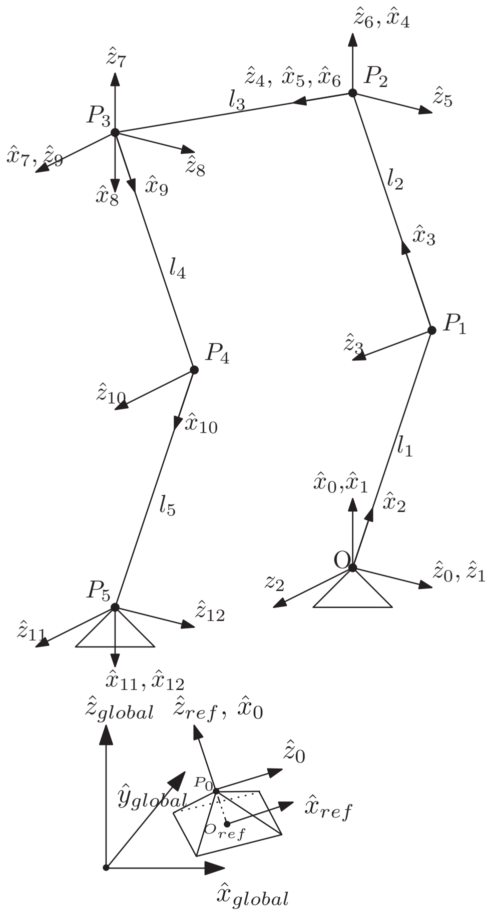


Fig. 3. Biped coordinate frames.

Furthermore, the position of the knee joint can be obtained by drawing two circles $\{C_1, C_2\}$ of radii $\{l_1, l_2\}$ at points {A(0, 0), C $(({}^1x_h, {}^1z_h))$}, respectively. Note that, ${}^1X_h = {}^{global}_{1}T\ {}^{global}X_h$. Of the two possible points of intersection obtained using the above formulation, the desired solution must ensure that the knee joint should be physically realizable, i.e., $\theta_3 > 0$. In order to attain this condition, only the point of intersection $({}^1x_k, {}^1z_k)$, satisfying the inequality, ${}^1z_k > {}^1z_h/2$ is accepted. Having obtained the position of the knee joint and taking into account the convention for representing joint angles, we have

$$\theta_2 = \text{atan2}(-{}^1z_k, {}^1x_k) \tag{6}$$

In order to determine the remaining joint angles, the following equation is used:

$${}^3_4T\ {}^4_5T\ {}^5_6T = {}^{global}_{3}T^{-1}\ {}^{global}_{6}T \tag{7}$$

Note that the right-hand side (RHS) of the above equation is known, say ${}^kT$ (the orientation of the hip joint is available from Eq. (15)), the angles $\theta_4, \theta_5, \theta_6$ can now be determined analytically:

$${}^3_4R\ {}^4_5R\ {}^5_6R = {}^kR = \begin{bmatrix} -c\theta_4 s\theta_5 c\theta_6 + s\theta_4 s\theta_6 & c\theta_4 s\theta_5 s\theta_6 & c\theta_4 c\theta_5 \\ -s\theta_4 \theta_5 c\theta_6 - c\theta_4 s\theta_6 & s\theta_4 s\theta_5 s\theta_6 - c\theta_4 c\theta_6 & s\theta_4 c\theta_5 \\ c\theta_5 c\theta_6 - s\theta_5 s\theta_6 & -c\theta_5 s\theta_6 + c\theta_6 & s\theta_5 \end{bmatrix} \tag{8}$$

Therefore,

$$\theta_5 = \sin^{-1}({}^kR(3, 3)), \qquad -pi/2 < \theta_5 < pi/2 \tag{9}$$

https://doi.org/10.1017/S0263574718000188 Published online by Cambridge University Press

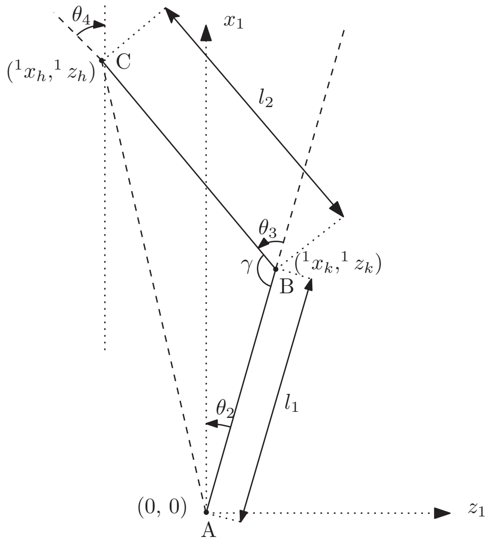


Fig. 4. Inverse kinematics in the sagittal plane.

$$\theta_4 = \text{atan2}({}^kR(2,3)/c\theta_5, {}^kR(1,3)/c\theta_5) \tag{10}$$

$$\theta_6 = \cos^{-1}({}^kR(3,1)) - \theta_5 \tag{11}$$

In a similar manner, inverse kinematics can also be performed for the other leg. As a result θ_i, $i \in [1, 12]$ are obtained.

3. Dynamic Modeling

The goal of inverse dynamic calculations is to determine the joint torques needed to generate a given motion. The Euler–Lagrange approach for robot manipulators[36] is used for this purpose:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = \varphi_i, \;\; i = 1, 2, \ldots, n \tag{12}$$

L = Lagrangian function = $E_k - E_p$ (Kinetic Energy − Potential Energy);
q_i = Generalized coordinate of the robot;
φ_i = Non-conservative generalized forces applied at joint i to drive link i.

In this discussion,

$$\varphi_i = J^T F_{c_i} + \tau_i$$

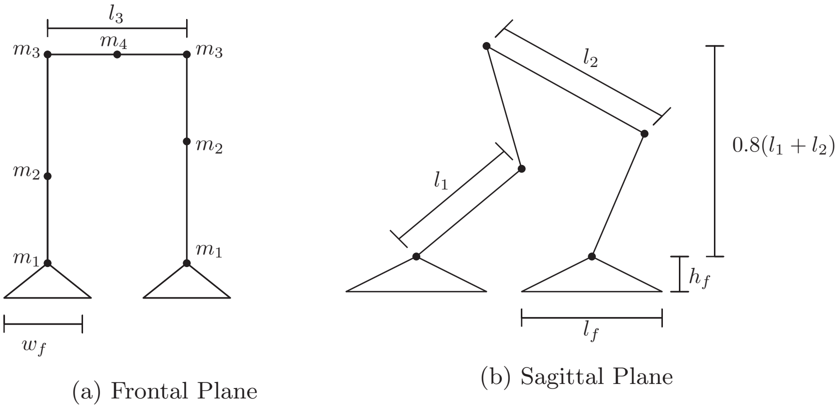
Here, $J^T F_{c_i}$ is the term due to external contact forces and τ_i is the actuator torque. All 12 DoF of the model are fully actuated.

External forces. In the single support phase (SSP), the swing foot is in the air and hence the ground reaction force (GRF) on the stance foot can be uniquely determined as $F_c = ma_{com} - mg$. During the double support phase (DSP), however, the system becomes indeterminate and only an approximate solution can be obtained. In the current study, the velocity of the swing foot reduces to zero just before it touches the ground, so do the velocities of all the actuators. Determination of the contact forces during the double support phase is a field of study in itself. Some authors optimize the stability margin and internal forces work while treating the contact forces as variables;[37] Dai *et al.*[38] framed the contact forces during the entire gait cycle as variables for the optimal control problem. In this work, an impact-less linear transition model has been utilized here, i.e., if the right foot is about to land on the ground, the force (f_r) on the right foot is modeled as $f_r = \lambda(ma_{com} - mg)$, where λ varies linearly from 0 to 1 from the start to end of DSP. More complex and evolved models can be used to determine

https://doi.org/10.1017/S0263574718000188 Published online by Cambridge University Press

Table III. Biped parameters.

m_1 (kg)	m_2 (kg)	m_3 (kg)	m_4 (kg)	l_1 (m)	l_2 (m)	l_3 (m)	h_f (m)	l_f (m)	w_f (m)
0.194	0.079	0.290	0.344	0.111	0.111	0.085	0.032	0.110	0.070



(a) Frontal Plane

(b) Sagittal Plane

Fig. 5. Biped parameters (a) Frontal plane. (b) Sagittal plane.

the impact forces during the double support phases, but they do not affect the underlying theme of this study, and they can simply be plugged in alongside the calculations for the system dynamics.

The standard form of the dynamic equation for rigid biped is derived as

$$D(\theta)\ddot{\theta} + B(\dot{\theta}, \theta) + G(\theta) - \tau_{ext} = \tau, \tag{13}$$

D is the 12×12 symmetric and positive-definite generalized mass matrix, the vector B regroups Coriolis and centrifugal terms, G represents gravity terms and τ is the joint torque vector. $\tau_{ext} = 0$ during SSP and $\tau_{ext} = J^T F_{c_{12}}$ during DSP where $F_{c_{12}}$ is the linearly modeled GRF on the final link (link 12) and J is the Jacobian. Work done for the rigid-link biped during gait is calculated as

$$\mathrm{W_R} = \sum_{i=1}^{12} \int_0^{t_c} |\tau_i.\dot{\theta}_i| \mathrm{d}t \tag{14}$$

Schematic of biped in frontal and sagittal views is presented in Fig. 5. A lumped mass model has been used to simulate the biped. The mass m_1 is fixed at the ankle joint, m_2 is concentrated at the knee joint, m_3 is located at the hip joint and m_4 represents the mass at mid-hip. The length of the shank is l_1, thigh is l_2, the hip to hip distance is l_3, and the length, width and height of the foot are represented by l_f, w_f and h_f, respectively. The mid-hip height was kept constant at $0.8(l_1 + l_2)$; this has been reported to be close to the optimal hip height during walking.[39] The time for each step was kept constant at 1s. The values of biped parameters are listed in Table III.

4. Trajectory Generation

In this section, a methodology for generating the trajectory for given initial and final states is presented; this is a *per-step* procedure. Given the initial and final states of the robot, the goal of trajectory generation is to generate smooth joint motion that results in a balanced and energetically optimal gait. The gait cycle has been proposed to begin in the middle of the DSP; in one step, the biped completes one full cycle and returns to the same phase; also, the gait is considered symmetric about the two legs. Trajectories have been generated for the swing foot and the mid-hip in the task space (Fig. 6) relative to a reference frame on the stance foot (Fig. 3).

- t_0: Hip starts moving forward and sideways, no movement of either feet.
- t_{d_1}: Swing foot starts moving, DSP ends, SSP begins.

https://doi.org/10.1017/S0263574718000188 Published online by Cambridge University Press

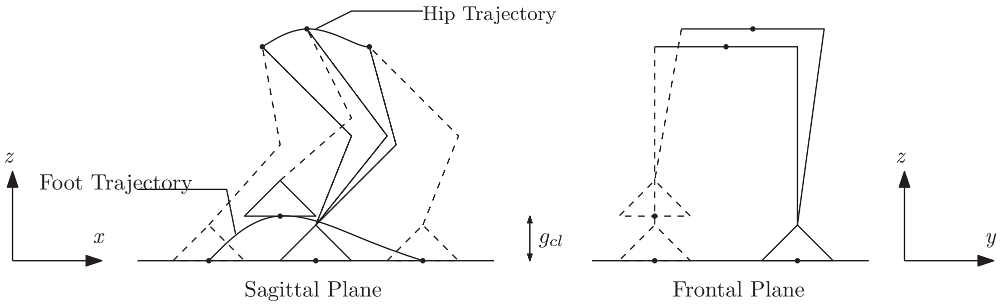


Fig. 6. Trajectory generation for foot and mid-hip.

- t_m: Midpoint of the gait cycle, swing foot is at maximum ground clearance.
- t_a: Mathematically expressed as $(t_m + t_{d_1})/2$, continuation of SSP.
- t_b: Expressed as $(t_m + t_{d_2})/2$, continuation of SSP, similar to t_a.
- t_{d_2}: Marks the end of SSP, swing foot stops moving.
- t_f: Biped comes to momentary rest, completion of single step cycle.

4.1. Cartesian coordinates

For planning all the trajectories, time is chosen to be the base variable, i.e., all trajectories will be generated in terms of time. A $(n_i + 6)$-order polynomial function is used for trajectory generation, where n is the number of intermediate points considered between the initial and final state; this is done to ensure that the first- and second-order derivatives vanish at end points. Time-dependent trajectories are generated in the x, y and z directions independently; the choice of intermediate points is also highlighted.

Swing foot trajectoy.

- x direction: The bulk of movement takes place in this direction; it necessitates the use of at least two control points; since the swing foot does not move in DSP, t_a and t_b are used for this purpose:

$$x_{sf_{t_a}} = x_{sf_{t_0}} + (f_1 - 0.5)(x_{sf_{t_f}} - x_{sf_{t_0}})$$
$$x_{sf_{t_b}} = x_{sf_{t_0}} + (1.5 - f_2)(x_{sf_{t_f}} - x_{sf_{t_0}})$$

  f_1 and f_2 are variables in [0, 1] to be determined by optimization. The quantities $\{f_1 - 0.5\}$ and $\{1.5 - f_2\}$ represent the fraction of total distance the swing foot has moved at times t_a and t_b, respectively.
- y direction: There is no change in the y-coordinate of swing foot during straight walk; even during turning, this change is very small; keeping this in consideration, the motion variation is kept linear in $\{t_{d_1}, t_{d_2}\}$.
- z direction: Ground clearance is needed for biped locomotion; this is represented by g_{cl} in Fig. 6 and the point of this clearance is t_m.

Mid-hip trajectory.

Mid-hip is the midpoint of the two hip-joints. The path followed by mid-hip is vital in determining the dynamic balance as well as energy consumption of biped during locomotion; this requires a higher number of variables to adequately define the motion.

- x direction: For walking on uneven terrain, dynamic balance requires a variety of hip motions. Pertaining to this requirement, two DoF to hip motion in a way similar to swing foot:

$$x_{mh_{t_a}} = x_{mh_{t_0}} + f_3(x_{mh_{t_f}} - x_{mh_{t_0}}),$$
$$x_{mh_{t_b}} = x_{mh_{t_0}} + f_4(x_{mh_{t_f}} - x_{mh_{t_0}})$$

https://doi.org/10.1017/S0263574718000188 Published online by Cambridge University Press

f_3 and f_4 are optimization variables in the closed interval [0,1] and they represent the fraction of total distance the mid-hip has moved at times t_a and t_b, respectively.

- y direction: Sideways tilt of mid-hip is required for maintaining lateral balance; it is also the single largest contributor to energy consumption, particularly for smaller step lengths. Intermediate points for polynomial interpolation are selected as t_{d_1}, t_{d_2} and t_m. t_{d1} and t_{d_2} mark the beginning and end of SSP, respectively, and t_m is used to ensure that the sideways tilt is not more than what is necessary as it affects the power consumption. The movement of mid-hip along the y direction is expressed in terms of initial positions of the stance and swing hips $(y_{st.h_{t_0}}, y_{sw.h_{t_0}})$.

  For a walking motion without turning, $y_{st.h_{t_0}}, y_{sw.h_{t_0}} = w_t$, where w_t is trunk width, i.e., if $f_5 = 1$, it would imply a sideways motion of mid-hip by an amount w_t at time t_m; it is equivalent to a tilt of almost 30°, which is much larger than the typical biped requirements. To avoid undesirable iterations of the optimization program, the values of f_5, f_6 and f_7 were confined in the interval [0.0, 0.5]:

$$\begin{aligned} y_{mh_{t_{d_1}}} &= y_{mh_{t_0}} + f_6(y_{st.h_{t_0}} - y_{sw.h_{t_0}}) \\ y_{mh_{t_m}} &= y_{mh_{t_0}} + f_5(y_{st.h_{t_0}} - y_{sw.h_{t_0}}) \\ y_{mh_{t_{d_2}}} &= y_{mh_{t_0}} + f_7(y_{st.h_{t_0}} - y_{sw.h_{t_0}}) \end{aligned}$$

- z direction: Vertical movement of mid-hip has a little effect on dynamic balance; this movement, however, is necessary for walking on uneven terrain in an energy-efficient manner. A single intermediate point t_m is used to define this path. Also used are the initial $(z_{mh_{t_0}})$ and final $(z_{mh_{t_f}})$ heights of mid-hip. $\Delta z_{mh_{max}}$ is the maximum allowable movement of the hip in vertical direction, this was set at 2.5 cm:

$$z_{mh_{t_m}} = max(z_{mh_{t_0}}, z_{mh_{t_f}}) + (f_8 - 0.5)\Delta z_{mh_{max}}$$

Trajectories for various step lengths, terrain slope and turning angles are generated as shown in Fig. 7. Points used for interpolation have been marked in black; it is these points that determine the trajectory; they depend on the gait variables (f_i) and will be determined using optimization in Section 5.

4.2. Orientations of feet and mid-hip

Cartesian coordinates of the reference frames located at the two feet and the mid-hip alone, however, are not sufficient to determine the state of a 12 DoF biped; orientations of the two feet and mid-hip are also required as inputs, and these orientations are provided in the form of Euler angles in a global reference frame.

Foot orientation. The foot orientations are kept such that the supporting surface remains parallel to the ground. In addition, turning is permitted as per the requirement. In Fig. 8(a), the pitch angle of the stance foot changes from θ_0 to $(\theta_0 - \theta_{slope})$, while the orientation of stance foot remains unchanged. Similarly, in Fig. 8(b), the yaw angle of swing foot changes from ψ_0 to $(\psi_0 + \alpha)$. In the current work, gradient is only considered in one direction; hence, the roll angle (ϕ) is always kept 0. If the orientation of the swing foot changes from $\{\phi, \theta, \psi\}$ to $\{\phi + \Delta\phi, \theta + \Delta\theta, \psi + \Delta\psi\}$ from time t_{d_1} to t_{d_2}, then these angles are varied linearly with time in the interval $[t_{d_1}, t_{d-2}]$.

Mid-hip orientation. Generally, a camera is mounted along on top of a biped for navigational purposes; it is then desired to keep the trunk vertically aligned throughout the motion. The mid-hip is oriented such that its x-axis $(\hat{x}_{mh})$ points along the direction joining the two hip joints (Fig. 8(a) and (b)). Any change in the hip-orientation during motion is also varied linearly with time. Together, these constraints result in an orientation of the mid-hip that depends only on the yaw angle; it is expressed as follows:

$$^{global}_{mh}R = \begin{bmatrix} \cos(\theta_{st} + \alpha/2) & -\sin(\theta_{st} + \alpha/2) & 0 \\ \sin(\theta_{st} + \alpha/2) & \cos(\theta_{st} + \alpha/2) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{15}$$

https://doi.org/10.1017/S0263574718000188 Published online by Cambridge University Press

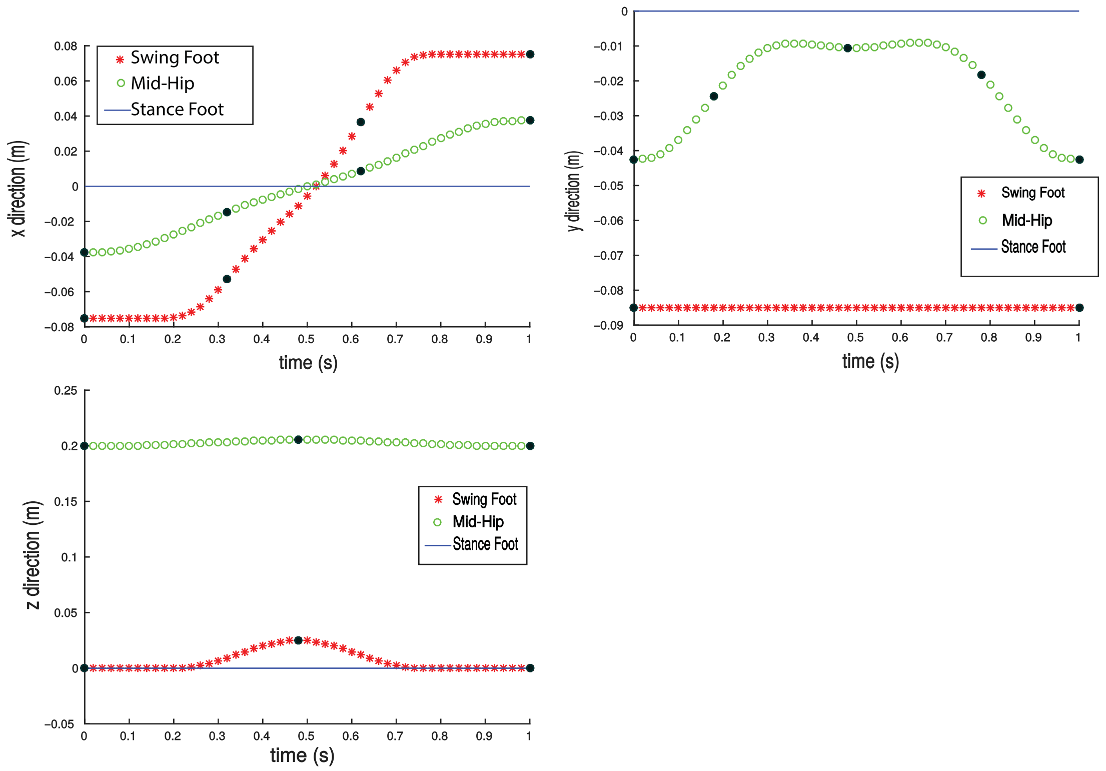
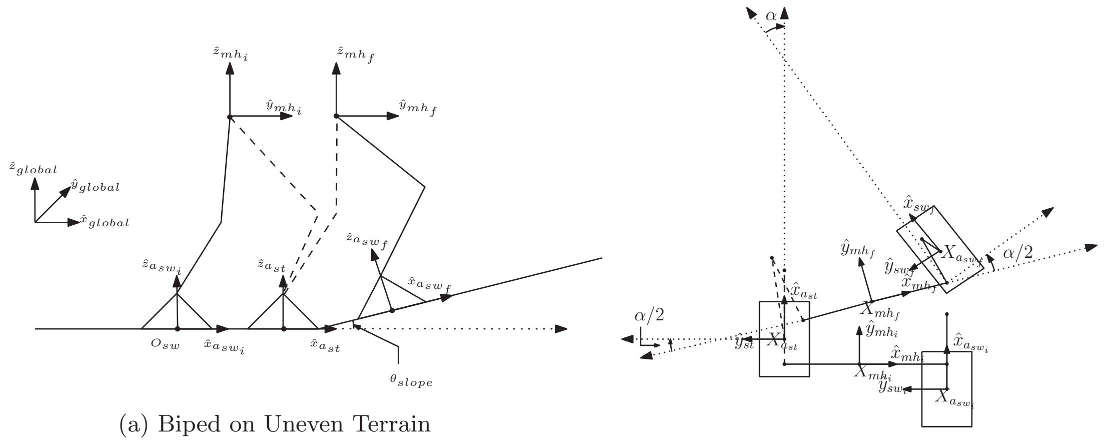


Fig. 7. Swing foot, stance foot and mid-hip trajectories in the x, y and z directions.



(a) Biped on Uneven Terrain

(b) Biped During Turning

Fig. 8. Biped movements. (a) Biped on uneven terrain. (b) Biped during turning.

5. Optimization Using GA

For performing the optimization, in addition to the energy cost (W_E), the balance of the biped is also taken into account. Different authors use different costs as a parameter to assess the quality of motion. Most commonly used methodologies involve minimizing the actuator work ($\int \tau\delta\theta$),[15] actuator torque ($\int \tau^2$)[40] or the mechanical cost of transport (E/mgd)[41] or a combination of these.[42,43] For a specified step length and step time (as is the case in this work), minimization of actuator work and the mechanical cost of transport are equivalent. In this study, actuator work is minimized to generate energy-efficient

https://doi.org/10.1017/S0263574718000188 Published online by Cambridge University Press

steps. Every walking step is discretized into 50 computational substeps for the purpose of ZMP and dynamic calculations. For any substep n, two calculations are performed:

$$W_{E_n} = \sum_{i=1}^{12} |\tau_n.\dot{\theta}_n| \Delta t \tag{16}$$

Here, $\Delta t = t_{step}/50$. A function $f(n)$ is defined as follows:

$$f(n) = \begin{cases} 0, & \text{if } zmp(n) \in \mathcal{SP} \\ 10, & \text{otherwise} \end{cases}$$

$\mathcal{SP}$ is the support polygon, and $zmp(n)$ is the location of zmp at substep n. Combining the two costs, the objective function to be minimized is C:

$$C = \sum_{n=1}^{50} (W_{E_n} + f(n)) \tag{17}$$

The optimization variables (x_i) are dimensionless for $i \in \{1, 2, 3, 4, 5, 6, 7, 8\}$. These are the same f_i as introduced in Section 4. Since the penalization on ZMP violation has been included in the objective function and since the range of variables f_i (x_i) was chosen appropriately in line with the kinematic constraints of the biped, no additional constraint is needed for the optimization.

Margin of stability. The supporting polygon $\mathcal{SP}$ for walking is the area of the supporting foot during the SSP and the convex hull of the two supporting feet during the DSP. At the time of optimization, however, a margin of stability is introduced as it provides a safety factor as shown in Fig. 9. $\mathcal{A}_1$ is the area of the support foot (SSP is considered), while $\mathcal{A}_2$ is the area obtained after taking the margin of stability into account; b was set to 1 cm. The dynamic equations of the biped robot (Eq. (13)) are 12 coupled non-linear equations and such optimization problems are well suited for GA. The inputs to the optimization problem were the dimensionless variables (f_i, $i = 1$ to 8) as described in Section 4.1 and the objective function was the energy consumption for each step, subject to the constraint of ZMP-based balance. The trajectories were generated for a variety of step lengths, slopes and turn angles. Optimization was performed for step lengths of $\{0.05,\ 0.10,\ 0.15,\ 0.20,\ 0.25\}$, terrain angles were $\{-10^\circ,\ -5^\circ,\ 0^\circ,\ 5^\circ,\ 10^\circ\}$ and turn angles were $\{-15^\circ,\ -5^\circ,\ 0^\circ,\ 5^\circ,\ 15^\circ\}$. Furthermore, it is often required to switch step lengths during walking; for this purpose, optimization was also performed from stand to walk and walk to stand transitions. These represent the extreme ends of transitions; all other transitions will lie somewhere in between. The concept is better illustrated in Fig. 10.

Figure 10(a) represents a standard walking case where swing foot moves a total distance of $2sl_0$. In this case, the swing foot is initial sl_0 distance behind the stance foot and then moves ahead by the same distance. Repeating this movement will yield a symmetric gait. The upper limit of sl_0 was set to be 0.125 m. Total computations for this case were $5 \times 5 = 25$ (for five different step lengths and five different slopes). Figure 10(b) represents a scenario where the biped is transitioning from standing pose to walking pose. Figure 10(c) represents the case when the transition is from walking pose to standing pose. Total computations for parts (b) and (c) were also 25 each. Figure 10(d) represents a general case where the transition is from some initial distance sl_1 to some final distance sl_2. Lower and upper bounds for both sl_1 and sl_2 are 0 and 0.125 m, respectively. The calculations for parts (a)–(c) were performed for all terrain angles. During non-zero turning angles $(-15^\circ\ -5^\circ,\ 5^\circ,\ 15^\circ)$, only case (a) was allowed; the computation was performed for $sl_0 \in \{0.025,\ 0.050,\ 0.075,\ 0.1,\ 0.125\}$ m and all the previously mentioned turn and terrain angles. It should also be noted that turning movement is not symmetric with respect to the two feet. During turning, the step length of the outer foot is more than the inner foot. Calculations were performed for both these movements resulting in total turning computations of $2 \times 5 \times 5 \times 2 = 100$. However, for a given turning angles θ and step length sl, the variables governing the movement of the right foot will be the same as the variables governing the movement of the left feet for a turning angle of $-\theta$.

https://doi.org/10.1017/S0263574718000188 Published online by Cambridge University Press

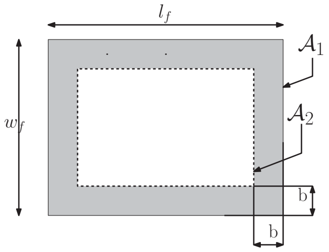


Fig. 9. Balance margin.

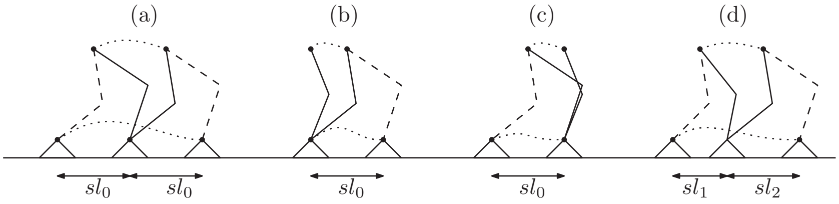


Fig. 10. Possible movements along straight line walk.

Taking into account movements of left and right foot, computation results for all the previously mentioned cases (a total of $175 \times 2 = 350$) were stored in a database. The results were the variables on which trajectory depended (f_i, $i \in \{1 \text{ to } 8\}$) and energy consumption (in Joules).

GA parameters. The population size was 50, the crossover probability was set to 0.9, mutation was 0.05 and the GA was terminated at $min(t_{lim}, gen_{lim})$, $t_{lim} = 5$ h, $gen_{lim} = 120$. Each variable (f_i) was represented with 6 bits, resulting in a total string length of $6 \times 8 = 48$.

High-performance computing. The computation was performed on a single node of Xeon E5-2670V 2.5 GHz 2 CPU-IvyBridge (20-cores per node) on HP-Proliant-SL-230s-Gen8 servers with 128 GB of RAM per node E5-2670v2x10 core 2.5 GHz and took close to 5 h for each run of the GA. Dynamic calculation of robot manipulators are $O(n^3)$ with respect to the number of DoF; for the presented system with 12 DoF, the calculation is computationally intensive. Parallelization of calculation resulted in a 20-fold increase in speed.

6. Neural Network for Trajectory Generation

GA-based optimization process for trajectory generation is time consuming; this makes the real-time implementation infeasible. To overcome this challenge, an artificial neural network ($\mathcal{NN}_1$) has been used for trajectory generation.

6.1. Feature identification

To identify a walking movement, it is reasonable to include the step length, turning angle, terrain slope and stepping foot (left or right) for trajectory generation. In addition to these, it is also necessary to determine the current state of the biped. In Fig. 10(d), the biped has a step length of $sl = sl_1 + sl_2$, whereas in Fig. 10(b), the step length is sl_0. Consider now a case when $sl_1 = sl_2 = sl_0/2$. This would imply that step lengths for parts (b) and (d) are the same, but clearly the two cases represent a very different type of movement. The dynamics governing the step as shown in (b) would be quite different compared to the case (d). Similar observation can also be made for case (c); keeping this in mind, the distance between the two feet before the execution of the step (sl_1) and the distance between the two feet after the step execution (sl_2) were identified as separate features. The features used for real-time trajectory generations were $\{sl_1,\ sl_2,\ m,\ sf,\ \alpha\}$. Here, m is the slope of the terrain represented by its value in degrees. sf is used to identify the swing foot, $sf = 1$ implies right foot swing and $sf = 0$ implies left foot swing and, α is the turn angle in degrees.

https://doi.org/10.1017/S0263574718000188 Published online by Cambridge University Press

Table IV. RMSE over test data using $\mathcal{NN}_1$.

f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
0.06	0.08	0.14	0.14	0.04	0.02	0.01	0.02

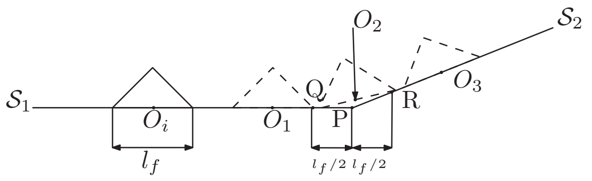


Fig. 11. Transition point.

6.2. Neural network architecture

A feed forward network with two hidden layers was used to model the non-linear mapping between the inputs $\{sf, sl_1, sl_2, m, \alpha\}$ and outputs $\{f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8\}$. The ANN consists of five input nodes, two hidden layers with 14 and 10 neurons each and eight output nodes. *tansig* activation function was used in the two hidden layers and the output layer was purely linear.

6.3. Data and performance

The total data of 350 input–output mappings was divided into training and test sets. In total, 85% of total data was randomly selected for training and the remaining was used to test and report the performance. Table IV represents the root mean square errors (RMSE) for each of the eight predicted variables reported with test data. The training was less than 6 s, while the prediction time was clse to 4 ms.

7. Step Planning

7.1. Navigation map and classification of obstacles

The step-planning program accepts a height map ($\mathcal{M}$) of the environment with a discretization, $dsc =$ 0.5 cm, along the x and y axes. Using $\mathcal{M}$, a navigation map ($\mathcal{NM}$) is generated. The map is scanned along the x and y directions to determine the gradients $\mathcal{M}_x$ and $\mathcal{M}_y$, respectively. Using the height and gradient information, the region is marked as *obstacle* or *free* in $\mathcal{NM}$. If the gradient along any direction is greater than 0.16 (10°), the grid point is marked as an obstacle. Also, a region of radius $r = sl_{max}/2$ (where $sl_{max} = 0.25$ m) around such a grid-point is also marked as *obstacle*; such a bounding cylinder approach provides a quick collision check at planning time. Collision check is performed for the center of this cylinder with the *obstacle* region. Furthermore, the gradients of $\mathcal{M}_x$ and $\mathcal{M}_y$ are determined and stored as $\mathcal{M}_{x_x}$ and $\mathcal{M}_{y_y}$.

The points with non-zero $\mathcal{M}_{x_x}$ and $\mathcal{M}_{y_y}$ values represent transition regions, i.e., the regions where there is a change in slope. While biped is allowed to traverse such areas, it cannot plant a step in this region. Consider Fig. 11, **P** represents a transition from surface $\mathcal{S}_1$ to surface $\mathcal{S}_2$. The initial position of the swing foot is denoted by O_i. The final positions O_1 and O_3 are acceptable as the support area of the foot lies on a single surface, O_2, however, must be discarded as surface contact cannot be established. While the actual area that must be avoided to ensure a single surface landing for the swing foot also depends on the biped orientation, in the pre-planning state, regions of length $l_f/2$ around the transition point are identified and marked separately in the map $\mathcal{NM}$ as *transition* (l_f is the length of the foot). At the time of step planning, a check is performed on the vertices of the landing foot to ensure that they all lie on the same supporting surface. Similar check will also be performed while crossing over *ditches*. A workspace environment comprising of a ramp and some obstacles is presented in Fig. 13. This represents the height map that is accepted as input to the map development program. The map is then discretized into individual cells,and classified as *obstacle* (black), *free* (white) or *transition* (green). After performing the cell classification, the map is obtained as shown in Fig. 14.

https://doi.org/10.1017/S0263574718000188 Published online by Cambridge University Press

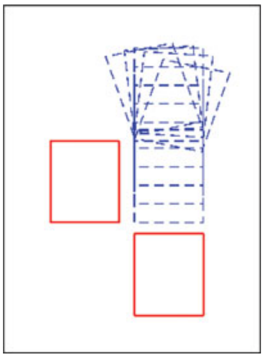


Fig. 12. Allowable foot transitions.

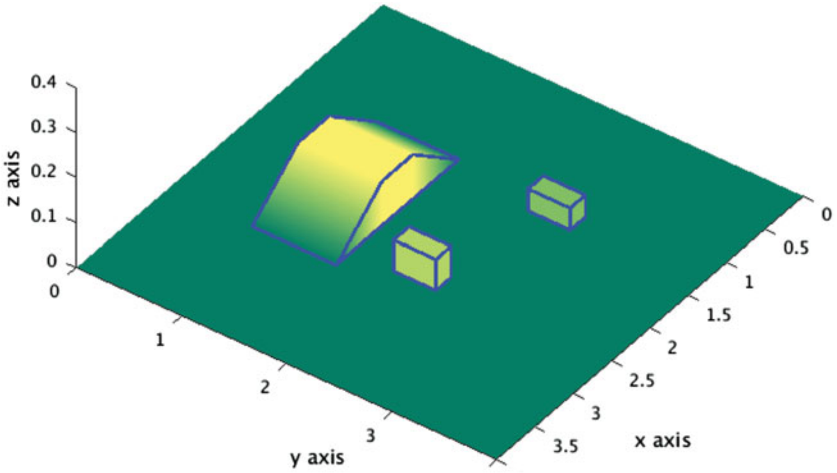


Fig. 13. Workspace for biped robot.

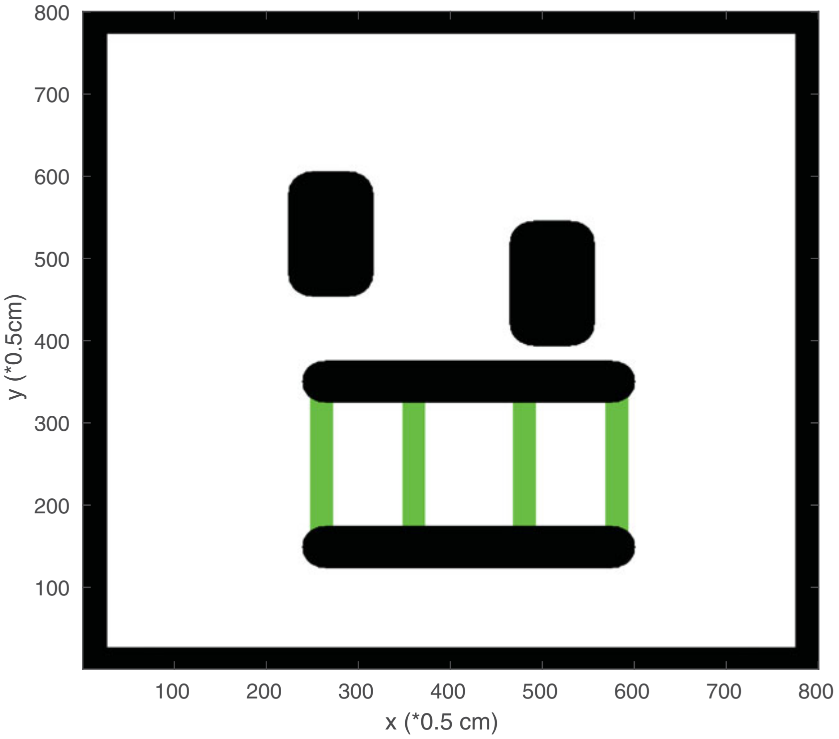


Fig. 14. Cell classification.

https://doi.org/10.1017/S0263574718000188 Published online by Cambridge University Press

7.2. State transitions

The problem of step planning is presented as a graph search problem where the aim is to reach from start state $\mathcal{S}_{start}$ to an end state $\mathcal{S}_{goal}$. Two states are said to be connected directly if the transition from state $\mathcal{S}_1$ to $\mathcal{S}_2$ can be accomplished using a single step. It is, therefore, desired to determine a sequence of actions ($\mathcal{A}_k, k \in \{1 \dots n\}$, where n is the total number of steps), which take the biped from a start position to the goal position (or within an acceptable distance, ϵ). The state ($\mathcal{S}$) of the biped for the purpose of step planning is determined by the positions and orientations of the two feet at any instant. The complete information regarding the biped can be encoded in a 12-D vector as follows:

$$S = [x_{lf},\ y_{lf},\ z_{lf},\ \phi_{lf},\ \theta_{lf},\ \psi_{lf},\ x_{rf},\ y_{rf},\ z_{rf},\ \phi_{rf},\ \theta_{rf},\ \psi_{rf}]$$

x and y represent the x and y coordinates of the foot, respectively, and z is the height from the ground, which is available from the height map, $z = \mathcal{M}(x, y)$. ϕ, θ, ψ are the Euler angles defining the orientation of the foot. In the current study, ϕ is taken to be zero in the free space, θ is determined using the gradient map $\mathcal{M}_x$, $\theta = \tan^{-1}(\mathcal{M}_x(x, y))$ and ψ represents the yaw angle of the foot, which is associated with turning motion. Subscripts lf, rf correspond to the left and right foot, respectively. The position $\mathcal{X}_{state}$ of any state is defined as the average of the positions of the two feet. $\mathcal{X}_{state} = [\frac{x_{lf}+x_{rf}}{2},\ \frac{y_{lf}+y_{rf}}{2}]$.

The transitions that can be accomplished from a given step in a single step are depicted in Fig. 12. The initial state of the two feet is displayed using solid lines (red); allowable transitions are displayed in dashed lines (blue). The state $\mathcal{S}_f$ of the biped after a transition is dependent on the current state $\mathcal{S}_c$, the terrain $\mathcal{T}$, the step length sl and turn angle $\Delta\psi$. The transition from one state to the next occurs in accordance with the kinematic model of the biped, the action $\mathcal{A}$ is the operator that takes the current state ($\mathcal{S}_c$), transition factors ($\mathcal{T}$, sl and $\Delta\psi$) as inputs and returns the final state:

$$\mathcal{S}_f = \mathcal{A}(\mathcal{S}_c,\ \mathcal{T},\ sl,\ \Delta\psi) \tag{18}$$

Alternatively, it can be stated that the operator $\mathcal{A}$ accepts $\{sf, sl_1, sl_2, m, \Delta\psi\}$ as inputs and returns the final state. The operator $\mathcal{A}$ has been described in 9. Every permutation of the five step parameters represents a unique transition t:

$$\mathcal{S}_f = \mathcal{A}(sf, sl_1, sl_2, m, \Delta\psi) \tag{19}$$

In order to maintain the repeatability and symmetry of the biped gait, change of step-length was not allowed during turning.

The allowable $sl_2 \in \{0,\ 0.025,\ 0.05,\ 0.075,\ 0.1,\ 0.125\}$ m and allowable turn angle $\Delta\psi \in \{-15^\circ,\ -5^\circ,\ 0^\circ,\ 5^\circ,\ 15^\circ\}$.

7.3. Cost of transport

Since the objective of the planner is develop an energetically efficient locomotion scheme, it is important to have an accurate estimate of *cost* of each step. While the exact cost for each action $\mathcal{A}$ can only be computed using dynamic calculation, the computation time it takes to perform that calculation severely limits their use in the planning process. Also, a pre-generation of all such costs is not possible as the slope of the allowable terrain represents a continuous spectrum from -10° to 10°. Besides, the combinations of other transition factors are also far too many to be computed in advance. Therefore, to develop an acceptable and accurate cost estimation system, a neural network was trained in a way similar to Section 6. The inputs to the network were the transition factors $\{sf, sl_1, sl_2, m, \Delta\psi\}$ and the output W was the energy consumption in Joules. A single hidden layer with 14 neurons was found to work well for energy estimation purpose; the RMSE on test-data (the same as Section 6.3) was 0.03 J. One major advantage of a trained network $\mathcal{NN}_2$ is that the need for heuristics to predict the energy cost is eliminated and real-time estimates of step cost are provided with acceptable accuracy. Therefore, the cost (C_i) of a transition t_i can be estimated as follows:

$$C_i = \mathcal{NN}_2(t_i) \tag{20}$$

https://doi.org/10.1017/S0263574718000188 Published online by Cambridge University Press

In order to guide the search in an A* search, the cost-to-go is defined using the current position and the goal position. Let the positions of the current and goal states be $\mathcal{X}_{current}$ and $\mathcal{X}_{goal}$, respectively. For this study, the maximum allowable step length sl_{max} is 0.25 m; each of sl_{1_0} and sl_{2_0} were, therefore, set to $sl_{max}/2$. The swing foot sf can be set to either 0 or 1 because there is no difference in energy consumption for straight walk as the motion is symmetric. The gradient m_0 is set to be $m_0 = (z_{goal} - z_{current})/|(\mathcal{X}_{goal} - \mathcal{X}_{current})|$; this is a very optimistic estimate of the cost-to-go and thus ensures optimality in A* search;[44] the turning angle $\Delta\psi_0$ is kept 0 to further ensure that the cost-to-go remains an underestimate of the actual cost. Since, the distance (D) between the current position and the goal position is $D = |\mathcal{X}_{current} - \mathcal{X}_{goal}|$; the minimum number of steps (n) required to reach the goal is $n = 2 * D/sl_{max}$. Therefore, the 'cost-to-go'($h(current, goal)$) is estimated as $h(current, goal) = n * \mathcal{NN}_2(1, sl_{1_0}, sl_{2_0}, m_0, 0)$.

7.4. Balance considerations

At the time of planning, it needs to be ensured that the step is dynamically executable. In Section 6, a neural network was trained using pre-generated data set to generate trajectories in real time. The same module is utilized in step planning; for every prospective step that lies in free space such that $sl_1 \geq 0.1$ m or $sl_2 \geq 0.1$ m, a secondary balance check is performed to ensure that the step is dynamically balanced. This check is only performed for longer step lengths as smaller steps are much easier to execute and do not require such a check.

7.5. Planning with A search*

The A* planner takes in as input – Start State ($\mathcal{S}_{start}$), a goal position ($\mathcal{S}_{goal}$), height map ($\mathcal{M}$), navigation map($\mathcal{NM}$), gradient maps $\mathcal{M}_x$, $\mathcal{M}_y$, the operator $\mathcal{A}$, set of all allowable transitions ($t_i,\ i \in 1, 2, \ldots, n$), trained ANNs ($\mathcal{NN}_1$, $\mathcal{NN}_2$) and a tolerance to the solution, $\epsilon = 2.5$ cm. If a step sequence is found, the planner returns an ordered list of transitions $t_i, i \in 1, 2, \ldots, k$ that lead the biped to the goal; failure is reported otherwise. The function *PlanSteps* (Algorithm 1) is presented; after performing the A* search, the sequence of transitions can be obtained to generate the path and execute the motion.

7.6. Bounded relaxation and optimality

The admissibility of the algorithm guarantees an optimal solution but it also means that a lot of nodes have to be expanded. Often some relaxations are provided to speed up the algorithm albeit with a compromise in optimality. Such relaxations can be bounded to guarantee that the solution would be no more than $(1 + \lambda)$ times the optima, where λ is the chosen bound. One such bounding technique presented in the results is a weighted A* search,[45] which uses the cost function:

$$f(n) = g(n) + (1 + \lambda)h(n)$$

The admissible as well as relaxed cases were used for step-planning. The output transitions $t_i, i \in \{1, 2 \ldots k\}$ were then fed to the trajectory generation system, which resulted in balanced energy-efficient locomotion.

8. Simulations and Discussions

8.1. Case I: Gradient avoidance

The environment is depicted in Fig. 15. The green pole denotes the starting position, and the red pole denotes the goal position. Initially, biped is oriented along the positive x-axis. A ramp is present in between the start and goal, and the line directly connecting these two positions passes through the ramp. For this environment, two step-planning approaches were compared. In the first approach, the objective was to minimize the distance traveled by the biped, and in the second approach, the objective was energy minimization. For energy minimization, A* search as well as weighted A* search was performed.

https://doi.org/10.1017/S0263574718000188 Published online by Cambridge University Press

```
Algorithm 1: PlanSteps
input  : $\mathcal{S}_{start}$, $\mathcal{S}_{goal}$, $\mathcal{M}$, $\mathcal{NM}$, $\mathcal{M}_x$, $\mathcal{M}_y$, $\mathcal{A}$, $t$, $\mathcal{NN}_1, \mathcal{NN}_2$, $\epsilon$
output: A sequence of transitions $t_i$ and parent states $\mathcal{S}_i$, $i \in 1, 2...k$ where $\mathcal{S}_k = \mathcal{S}_{goal}$
closedSet := { } ;
openSet := { $\mathcal{S}_{start}$ } ;
transitions := { }, parent : = { } ;
gScore := map with default value of Infinity ;
gScore{$\mathcal{S}_{start}$} := 0;
fScore := map with default value of Infinity ;
fScore{$\mathcal{S}_{start}$} := $h(start, goal)$;
while openSet in not emply do
    current: = the node in openSet having the lowest fScore value ;
    if $|\mathcal{X}_{current} - \mathcal{X}_{goal}| < \epsilon$ then
        return transitions
    end
    openSet.Remove(current);
    closedSet.Add(current);
    for $i$ = 1 to $n$ do
        if $\mathcal{X}_{neighbor} \in$ obstacle OR TransitionCheck($\mathcal{X}_{neighbor}$) FAILS then
            continue;
        else if BalanceCheck($\mathcal{X}_{current}$, $X_{neighbor}$)FAILS then
            continue;
        end
        neighbor = $\mathcal{A}(t_i)$ ;
        // Ignore the neighbor which is already evaluated
        if neighbor in ClosedSet then
            continue
        end
        gScore$_{temp}$ := gScore{$\mathcal{S}_{current}$} + $\mathcal{N}(t_i)$ ;
        // Discover a new node
        if neighbor not in openSet then
            openSet.Add(neighbor)
        end
        // This is not a better path
        if $gScore_{temp} \geq gScore\{neighbor\}$ then
            continue
        end
        // transit factors stored
        transitions{ $\mathcal{S}_{current}$} = $t_i$ ;
        parent{$\mathcal{S}_{neighbour}$} = $\mathcal{S}_{current}$ ;
        gScore{neighbour} = gScore$_{temp}$ ;
        fScore{neighbour} := gScore{neighbour} + $h(neighbour, goal)$ ;
    end
    return failure ;
end
```

- *Distance minimization*: The cost of each step ($g = |\mathcal{X}_{next} - \mathcal{X}_{current}|$) is the distance traveled by the biped; similarly, the heuristic guiding the search is the distance between the current position and the goal position ($h = |\mathcal{X}_{goal} - \mathcal{X}_{current}|$). The progression of graph search is presented in Fig. 16(a).

  The pink pixel denotes the starting position, the yellow pixel is the goal position, the green area represents the *transition* region, the black area defines the *obstacles*, red pixels denote the open nodes and blue pixels show the closed nodes.

https://doi.org/10.1017/S0263574718000188 Published online by Cambridge University Press

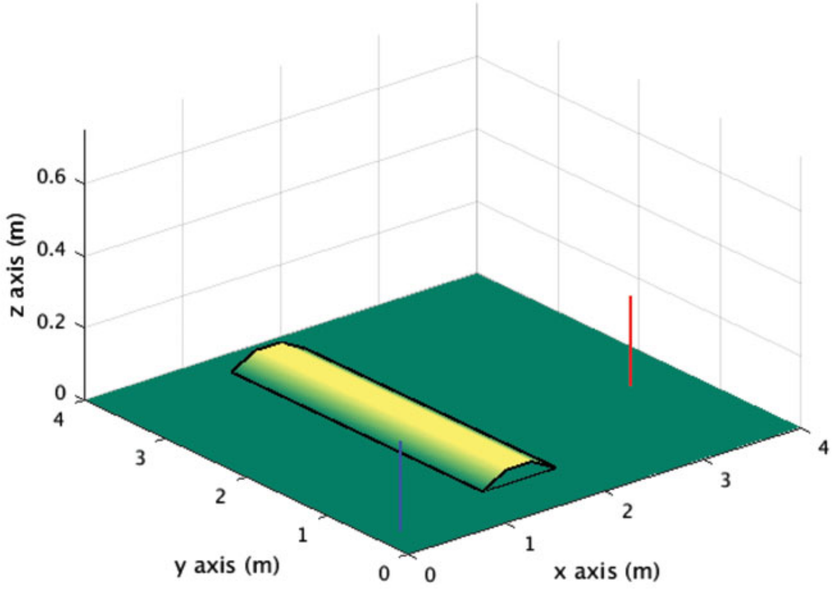


Fig. 15. Case I: Biped environment.

- *Energy minimization*: Algorithm 1 has been implemented for energy minimization. Two cases of bounded optimality have been presented. λ values of 0 and 0.15 were chosen for comparison. Figure 16(b) and (c) shows the graph progression for these cases and analysis is presented thereafter.

Analysis of graph search progression

- *Distance minimiztion*: In Fig. 16(a), the node expansion is very sparse; it can be attributed to the fact that in the case of distance minimization, the best possible path is a straight line connecting the start and goal positions as no obstacle is present in between; the graph search returns as straight a line as allowable due to the kinematic constraints of the biped.
- *Energy minimization,* $\lambda = 0.00$: A very dense exploration in Fig. 16(b) is due to the fact that the search heuristic is an underestimate of the true minima cost-to-go; the penalty for a guaranteed optima is the computation time. The search also explores the two options (over the ramp and on the side of the ramp) and selects the most efficient path.
- *Energy minimization,* $\lambda = 0.15$: In Fig. 16(c), graph exploration is sparse; this is because the search heuristic is a slight overestimate of the actual cost to go. The search progresses rapidly toward the goal position while successfully avoiding the gradient.

The results of the step plans are available in Fig. 17(a) and (b). Black steps refer to plan generated by distance minimization, yellow represents the energy minimization with $\lambda = 0.0$ and red color represents energy minimization with $\lambda = 0.15$. Performances of the resulting step plans are noted in Table V. $E_{\mathcal{NN}_2}$ represents the energy as computed by $\mathcal{NN}_2$ directly, whereas $\mathcal{NN}_1$ is the energy as obtained by the dynamics calculations on the results of $\mathcal{NN}_2$.

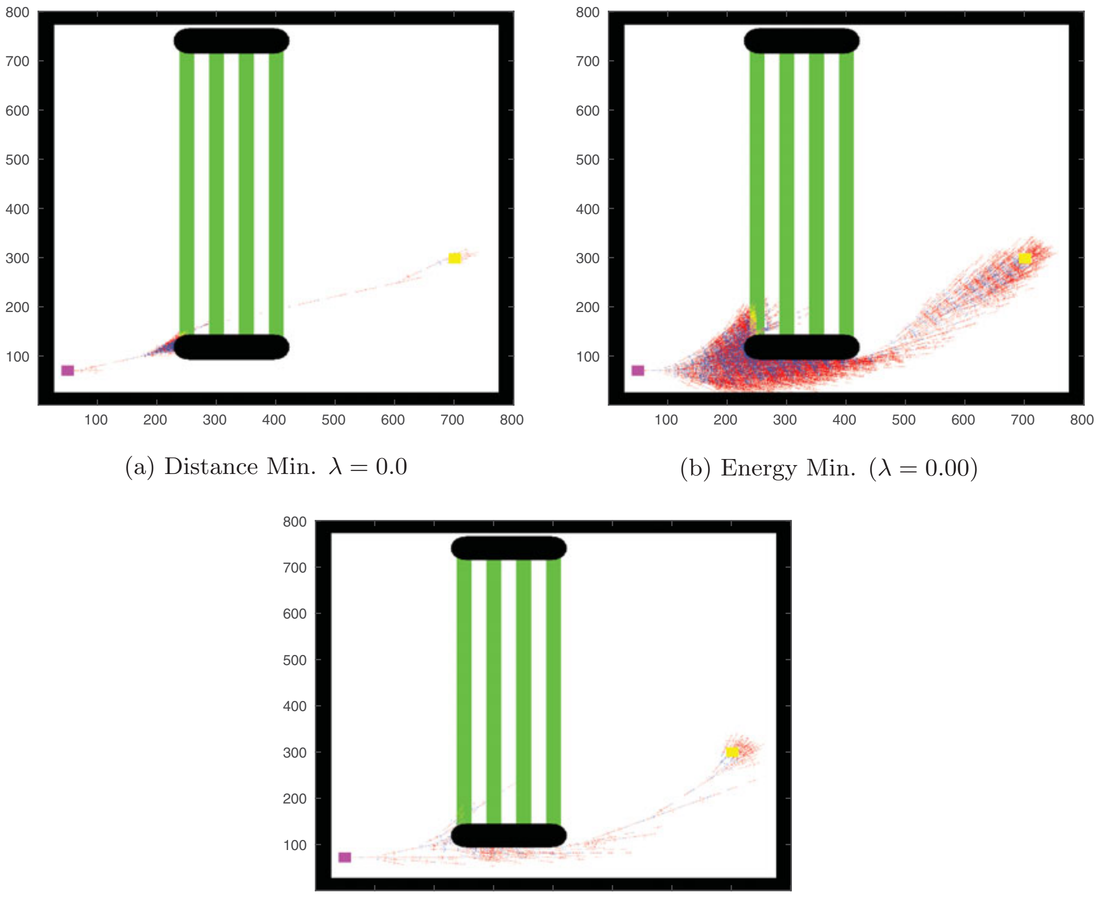
The step plan from distance minimization yielded global minima subject to the discretization and kinematic constraints of the biped. The path, however, traversed some slope prior to reaching the goal; to traverse changing slope, it also required a large number of smaller steps. Planning with the objective of energy minimization was able to avoid the ramp albeit with a slightly longer path. It should, however, be noted that the energy consumption for the two different λ values is comparable.

Optimality and computation. Had the heuristic function guiding the search been able to accurately estimate the cost-to-go, the solution provided by a weighted A* search would be *at worst* λ times more *expensive* than the optimal solution. In the presented approach, however, the heuristic guiding the search is, in fact, a conservative underestimate of the cost-to-go; this means that the resulting solution with weighted A* search would in fact be *better* than the mathematical guarantee. The extent to which the solution provided by a weighted search is closer to the true optima will vary from problem to problem. In general, the more complex the environment is, the closer is the solution of weighted

https://doi.org/10.1017/S0263574718000188 Published online by Cambridge University Press

Table V. Case I: Comparison of step-planning techniques.

Minimization	Weight$(1+\lambda)$	$E_{\mathcal{NN}_2}$ (J)	$E_{\mathcal{NN}_1}$ (J)	Movement (m)	Color
Distance	1.0	40.936	42.85	3.45	Black
Energy	1.15	34.244	35.43	3.59	Red
Energy	1.0	33.397	34.02	3.57	Yellow



(a) Distance Min. $\lambda = 0.0$

(b) Energy Min. $(\lambda = 0.00)$

(c) Energy Min. $(\lambda = 0.15)$

Fig. 16. Case I: Graph search progression (a) Distance min. $\lambda = 0.0$. (b) Energy min. $(\lambda = 0.00)$. (c) Energy min. $(\lambda = 0.15)$.

search to the true optima. In the presented case, the solution provided by $\lambda = 0.15$ is very close to the true optima, i.e., $\lambda = 0$ and it consumed only a fraction of the time for computation. Henceforth, energy minimization will only be presented with $\lambda = 0.15$.

8.2. Case II: navigating uneven terrain

The environment is depicted in Fig. 18. The green pole denotes the starting position, and the red pole denotes the goal position. Initially, biped was oriented along the positive x-axis. Start and goal positions are separated by an uneven surface. Similar to the previously discussed case, two step-planning approaches were compared – distance minimization and energy minimization. For energy minimization, weighted A* search was performed with $\lambda = 0.15$.

Step plans obtained as a result of the graph searches are depicted in Fig. 19 and Table VI compares the performance of the two approaches.

As expected, the approach of energy minimization resulted in a plan that consumes less energy than distance minimization approach.

https://doi.org/10.1017/S0263574718000188 Published online by Cambridge University Press

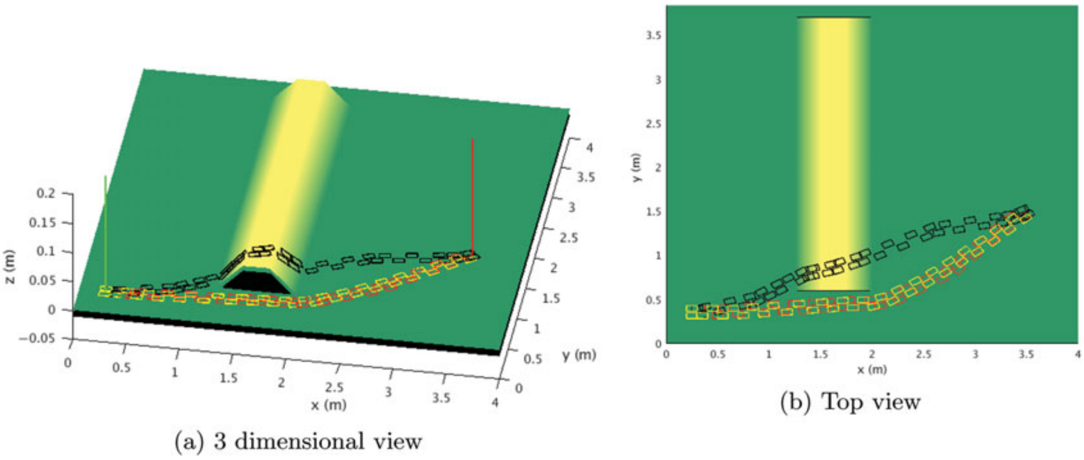


Fig. 17. Case I: Foot-step plans (a) 3-dimensional view. (b) Top view.

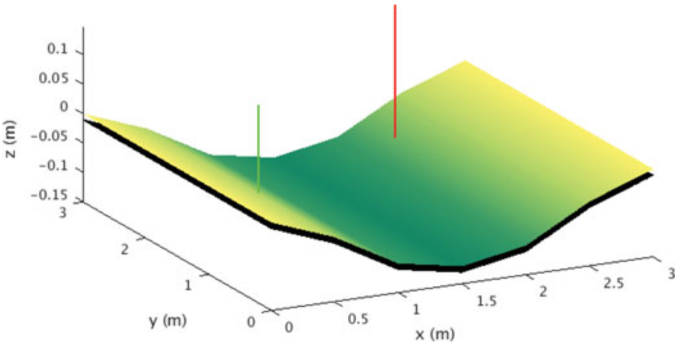


Fig. 18. Case II: Biped environment.

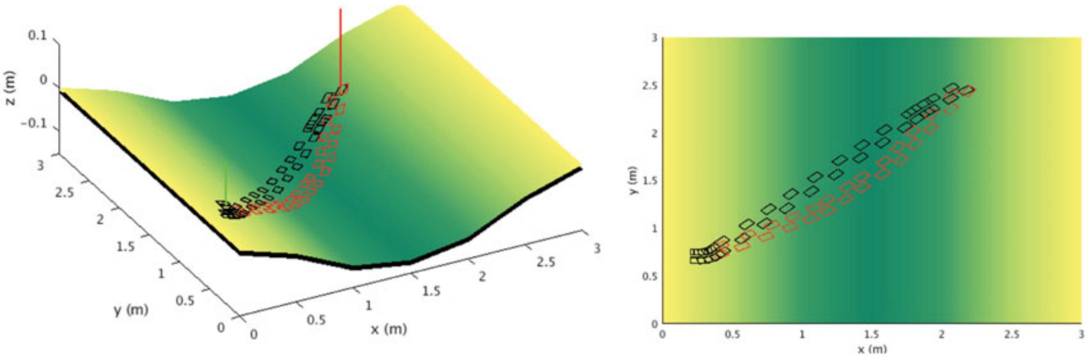


Fig. 19. Case II: Foot-step plans.

8.3. Case III: Ditch crossing

The environment is presented in Fig. 20; the white region represents ditch locations. While developing the navigation map, cells of the region around this obstacle are marked as *transition*; whenever the foot is in this region, its placement is checked to ensure that it lies *entirely* on the free surface and not on the ditch. To ensure placement, in addition to the four vertices, midpoints of the sides of foot are also checked to lie on the supporting surface. The developed footstep plan with the objective of energy minimization and $\lambda = 0.15$ is presented in Fig. 21.

https://doi.org/10.1017/S0263574718000188 Published online by Cambridge University Press

Table VI. Case II: Comparison of step-planning techniques.

Minimization	Weight $(1+\lambda)$	$E_{\mathcal{NN}_2}$	Movement (m)	Step color
Distance	1.0	31.33	2.65	Black
Energy	1.15	29.94	2.73	Red

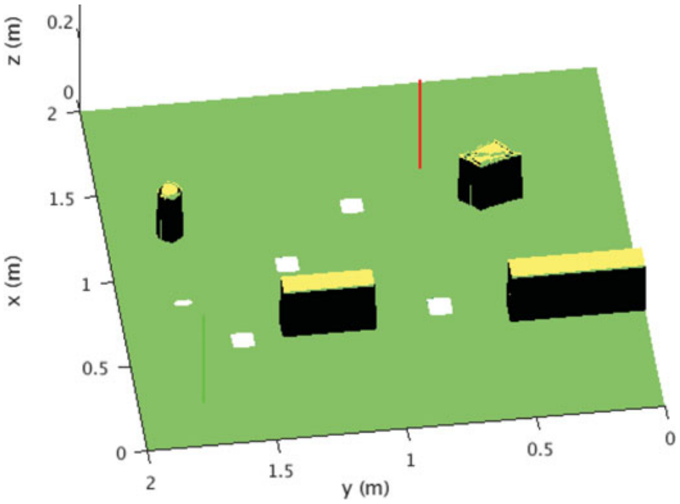


Fig. 20. Case III: Biped environment.

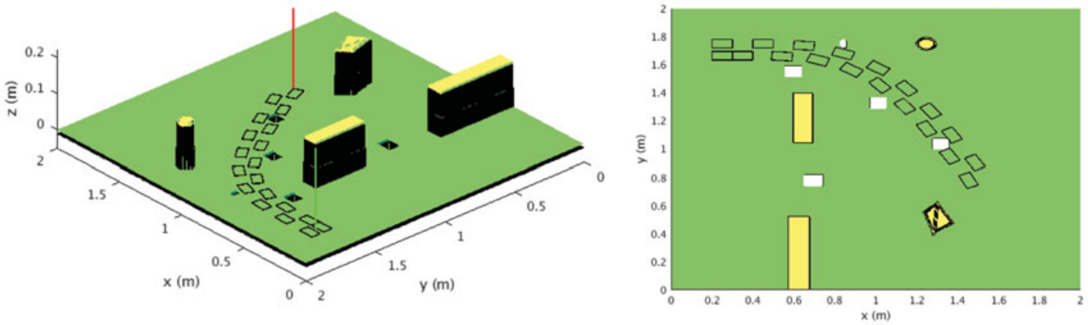


Fig. 21. Case III: Foot-step plans.

9. Conclusion and Future Work

In this study, energy-efficient locomotion scheme was developed for 12 DoF biped robot for walk on uneven terrain. Kinematic and dynamic modeling of the biped was performed to minimize the per-step energy consumption of the robot for a variety of step parameters like step length, terrain slope and turn angle while considering balance and kinematic constraints. A database containing the gait variables and the energy consumption for various step parameters was utilized to train two neural networks – one for real-time (5 ms) trajectory generation and another for accurate estimation of energy consumption. The proposed planning algorithm was implemented in MATLAB and was able to generate a sequence of 30–50 steps on an uneven surface in a few minutes. Finally, the trajectory generation module was integrated with the step planner to execute the biped motion. This approach guarantees step plan, which can be executed in a balanced manner, subject to the level of discretization in the A* planner. The programs associated with the work are available on github (https://github.com/ggupta9777/Energator).

Main contributions of this work can be summarized as follows:

- *Dynamic considerations in step planning*: The proposed scheme considers biped robot dynamics in step planning, which very few researchers have taken into account. This approach guarantees step

https://doi.org/10.1017/S0263574718000188 Published online by Cambridge University Press

plan, which can be executed in a balanced manner, if one exists, subject to the level of discretization in the A* planner.

- *State transition model*: The state of the biped was defined in terms of the positions and orientations of the two feet and mid-hip. Given an initial state, the transition model determined the final state using simple inputs such as step length, terrain map and turning angle. Trajectory generation between any two successive states was accomplished using a trained neural network in real time.
- *Energy efficiency in locomotion*: Most researchers have developed step plans without considering energy consumption; a few researchers who have paid attention to energetics have used overly simplistic models without explicit dynamic considerations. To the best of our knowledge, this is the first work that relies on accurate system model for energy estimation during step planning as well as trajectory generation.

With the proposed approach, step planning as well as trajectory generation was accomplished in diverse terrains. The framework integrated several aspects of biped locomotion to generate energy-efficient step plan and trajectory generation module in 3-D environment while considering system kinematics, dynamics and balance. A major advantage of the proposed scheme is that the need for approximate search heuristics in step planning is eliminated; also, the balance check and trajectory generation module ensure that the obtained step plan is executable. Last, the energy estimates are obtained using the dynamic model of the biped; this assures the energy efficiency in locomotion. Most researchers, thus far, have largely focussed on either step planning or gait generation, largely on planar environments. Some major challenges still need to be tackled; currently, step planning was accomplished in time anywhere between a few seconds to minutes; it was observed to increase drastically in congested areas and over long-range distances. To overcome that, the possibility of a tiered planning strategy needs to be explored. This could involve having a roadmap based approach at a higher level for course planning and a lower level A* search. Another exciting pursuit would be to develop a framework that accepts an adaptive set of step transitions and not just pre-defined primitives. The finite set of primitives puts an upper limit to the resolution of the search, minor adjustments in the step lengths and(or) turning angles can achieve superior performance and optimization.

References

1. M. Vukobratović and B. Borovac, "Zero-moment point – Thirty five years of its life," *Int. J. Humanoid Robot.* **1**(01), 157–173 (2004).
2. P. Sardain and G. Bessonnet, "Forces acting on a biped robot. Center of pressure-zero moment point," *IEEE Trans. Syst. Man Cybernet. Part A: Syst. Hum.* **34**(5), 630–637 (2004).
3. Q. Huang, K. Yokoi, S. Kajita, K. Kaneko, H. Arai, N. Koyachi and K. Tanie, "Planning walking patterns for a biped robot,"*IEEE Trans. Robot. Autom.* **17**(3), 280–289, (2001).
4. S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi and H. Hirukawa, "The 3D Linear Inverted Pendulum Mode: A Simple Modeling for A Biped Walking Pattern Generation," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1 (2001) pp. 239–246.
5. I.-W. Park, J.-Y. Kim, J. Lee and J.-H. Oh, "Online Free Walking Trajectory Generation for Biped Humanoid Robot KHR-3 (HUBO)," *Proceedings of the IEEE International Conference on Robotics and Automation ICRA2006* (2006) pp. 1231–1236.
6. J. Vermeulen, B. Verrelst, B. Vanderborght, D. Lefeber and P. Guillaume, "Trajectory planning for the walking biped Lucy," *Int. J. Robot. Res.* **25**(9), 867–887 (2006).
7. W. Gao, Z. Jia and C. Fu, "Increase the feasible step region of biped robots through active vertical flexion and extension motions," *Robotica* **35**(7), 1541–1561 (2017).
8. A. Takanishi, T. Takeya, H. Karaki and I. Kato, "A Control Method for Dynamic Biped Walking Under Unknown External Force," *Proceedings of the IEEE International Workshop on Intelligent Robots and Systems IROS'90, Towards a New Frontier of Applications* (1990) pp. 795–801.
9. P. R. Vundavilli and D. K. Pratihar, "Soft computing-based gait planners for a dynamically balanced biped robot negotiating sloping surfaces," *Appl. Soft Comput.* **9**(1), 191–208 (2009).
10. I. R. Manchester, U. Mettin, F. Iida and R. Tedrake, "Stable dynamic walking over uneven terrain," *Int. J. Robot. Res.* **30**(3), 265–279 (2011).
11. S. Bi, Z.-J. Zhuang, T. Xia, H.-X. Mo, H.-Q. Min and R.-H. Luo, "Multi-objective Optimization for a Humanoid Robot Walking on Slopes," *Proceedings of the International Conference on Machine Learning and Cybernetics ICMLC2011*, vol. 3 (2011) pp. 1261–1267.
12. W. Huang, C.-M. Chew, Y. Zheng and G.-S. Hong, "Pattern Generation for Bipedal Walking on Slopes and Stairs," *Proceedings of the 8th IEEE-RAS International Conference on Humanoid Robots Humanoids2008* (2008) pp. 205–210.

https://doi.org/10.1017/S0263574718000188 Published online by Cambridge University Press

13. Y. Zheng, M. C. Lin, D. Manocha, A. H. Adiwahono and C.-M. Chew, "A Walking Pattern Generator for Biped Robots on Uneven Terrains," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems IROS2010* (2010) pp. 4483–4488.
14. T. Arakawa and T. Fukuda, "Natural Motion Generation of Biped Locomotion Robot Using Hierarchical Trajectory Generation Method Consisting of GA, EP Layers," *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 1 (1997) pp. 211–216.
15. A. Sarkar and A. Dutta, "8-DoF biped robot with compliant-links," *Robot. Autonom. Syst.* **63**, 57–67 (2015).
16. I.-s. Lim, O. Kwon and J. H. Park, "Gait optimization of biped robots based on human motion analysis," *Robot. Autonom. Syst.* **62**(2), 229–240 (2014).
17. S. L. Cardenas-Maciel, O. Castillo and L. T. Aguilar, "Generation of walking periodic motions for a biped robot via genetic algorithms," *Appl. Soft Comput.* **11**(8), 5306–5314 (2011).
18. D. Gong, J. Yan and G. Zuo, "A review of gait optimization based on evolutionary computation," *Appl. Computat. Intell. Soft Comput.*, vol. 2010 (2010) p. 12.
19. G. Capi, Y. Nasu, L. Barolli, M. Yamano, K. Mitobe and K. Takeda, "A Neural Network Implementation of Biped Robot Optimal Gait During Walking Generated by Genetic Algorithm," *Proceedings of the Mediterranean Conference on Control and Automation*, Dubrovnik, Croatia (2001).
20. J. Nagasue, Y. Konishi, N. Araki, T. Sato and H. Ishigaki, "Slope-walking of a Biped Robot with k Nearest Neighbor Method," *Proceedings of the 4th International Conference on Innovative Computing, Information and Control ICICIC2009* (2009) pp. 173–176.
21. J. P. Ferreira, M. M. Crisostomo and A. P. Coimbra, "SVR versus neural-fuzzy network controllers for the sagittal balance of a biped robot," *IEEE Trans. Neural Netw.* **20**(12), 1885–1897 (2009).
22. P. R. Vundavilli and D. K. Pratihar, "Dynamically balanced optimal gaits of a ditch-crossing biped robot," *Robot. Autonom. Syst.* **58**(4), 349–361 (2010).
23. Z. Liu, L. Wang, Y. Zhang and C. P. Chen, "A SVM controller for the stable walking of biped robots based on small sample sizes," *Appl. Soft Comput.* **38**, 738–753 (2016).
24. J. Chestnutt, J. Kuffner, K. Nishiwaki and S. Kagami, "Planning Biped Navigation Strategies in Complex Environments," *Proceedings of the IEEE International Conference on Humanoid Robots*, Munich, Germany (2003).
25. J. Chestnutt and J. J. Kuffner, "A Tiered Planning Strategy for Biped Navigation," *Proceedings of the 4th IEEE/RAS International Conference on Humanoid Robots*, vol. 1 (2004) pp. 422–436.
26. W. Huang, J. Kim and C. G. Atkeson, "Energy-based Optimal Step Planning for Humanoids," *Proceedings of the IEEE International Conference on Robotics and Automation ICRA2013* (2013) pp. 3124–3129.
27. A. Hornung, D. Maier and M. Bennewitz, "Search-based Footstep Planning," *Proceedings of the ICRA Workshop on Progress and Open Problems in Motion Planning and Navigation for Humanoids* (Karlsruhe, Germany, 2013).
28. T.-Y. Li, P.-F. Chen and P.-Z. Huang, "Motion Planning for Humanoid Walking in a Layered Environment," *Proceedings of the IEEE International Conference on Robotics and Automation ICRA2003*, vol. 3 (2003) pp. 3421–3427.
29. R. Cupec, I. Aleksi and G. Schmidt, "Step sequence planning for a biped robot by means of a cylindrical shape model and a high-resolution 2.5 D map," *Robot. Autonom. Syst.* **59**(2), 84–100 (2011).
30. J.-S. Gutmann, M. Fukuchi and M. Fujita, "Real-time Path Planning for Humanoid Robot Navigation," *Proceedings of the International Joint Conference on Artificial Intelligence*, vol. 19 (Lawrence Erlbaum Associates Ltd., 2005) p. 1232.
31. S. Candido, Y.-T. Kim and S. Hutchinson, "An Improved Hierarchical Motion Planner for Humanoid Robots," *Proceedings of the 8th IEEE-RAS International Conference on Humanoid Robots Humanoids2008* (2008) pp. 654–661.
32. H. Prasanth and A. Sudheer, "A hybrid approach to motion planning for stepping over and obstacle avoidance in humanoids," *Proceedings of the CAD/CAM, Robotics and Factories of the Future* (Springer, 2016) pp. 651–664.
33. S.-H. Ryu, Y. Kang, S.-J. Kim, K. Lee, B.-J. You and N. L. Doh, "Humanoid path planning from hri perspective: A scalable approach via waypoints with a time index," *IEEE Trans. Cybernet.* **43**(1), 217–229 (2013).
34. N. Perrin, O. Stasse, L. Baudouin, F. Lamiraux and E. Yoshida, "Fast humanoid robot collision-free footstep planning using swept volume approximations," *IEEE Trans. Cybernet.* **28**(2), 427–439 (2012).
35. J. J. Craig, *Introduction to Robotics: Mechanics and Control*, vol. 3 (Pearson/Prentice Hall, Upper Saddle River, 2005).
36. K. S. Fu, R. Gonzalez and C. G. Lee, *Robotics: Control, Sensing, Vision, and Intelligence* (Tata McGraw-Hill Education, Singapore, 1988).
37. A. Dasgupta and Y. Nakamura, "Making Feasible Walking Motion of Humanoid Robots from Human Motion Capture Data," *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2 (1999) pp. 1044–1049.
38. H. Dai, A. Valenzuela and R. Tedrake, "Whole-body Motion Planning with Centroidal Dynamics and Full Kinematics," *Proceedings of the 14th IEEE-RAS International Conference on Humanoid Robots Humanoids2014* (2014) pp. 295–302.

https://doi.org/10.1017/S0263574718000188 Published online by Cambridge University Press

39. F. M. Silva and J. T. Machado, "Energy Analysis During Biped Walking," *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 1 (1999) pp. 59–64.
40. K. H. Koch, K. Mombaur and P. Soueres, "Optimization-based walking generation for humanoid robot," *IFAC Proc. Vol.* **45**(22), 498–504 (2012).
41. M. Srinivasan and A. Ruina, "Computer optimization of a minimal biped model discovers walking and running," *Nature* **439**(7072), 72–75 (2006).
42. C. Chevallereau and Y. Aoustin, "Optimal reference trajectories for walking and running of a biped robot," *Robotica* **19**(5), 557–569 (2001).
43. G. Capi, Y. Nasu, L. Barolli, K. Mitobe and K. Takeda, "Application of genetic algorithms for biped robot gait synthesis optimization during walking and going up-stairs," *Adv. Robot.* **15**(6), 675–694 (2001).
44. P. E. Hart, N. J. Nilsson and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybernet.* **4**(2), 100–107 (1968).
45. I. Pohl, "The Avoidance of (Relative) Catastrophe, Heuristic Competence, Genuine Dynamic Weighting and Computational Issues in Heuristic Problem Solving," *Proceedings of the 3rd International Joint Conference on Artificial intelligence* (Morgan Kaufmann Publishers Inc., 1973) pp. 12–17.

Appendix A

The methodology behind the determination of final state for a given initial state using simple step parameters like step-length (sl) and turn angle (α) is presented here. The cases with $\alpha = 0$ and $\alpha \neq 0$ will be discussed separately.

A.1. Transition without turning ($\alpha = 0$)

Given an initial state ($\mathcal{S}_i$) of the biped in any general configuration in the environment, if the turning angle is 0, the final state ($\mathcal{S}_f$) is only a function of step length sl. Instead of giving step-length sl as input, the desired distance between the two feet along the direction of movement sl_2 in the final state is provided as input (Fig. 22(a)). Due to kinematic and balance constraints, the upper bound of sl_1 and sl_2 is 0.125 m. In Fig. 22(b), biped moves from slope θ to slope 0 (or any other slope). A tentative position P_f' is evaluated along the plane of stance foot and its projection on the new plane (in this case, the plane with slope 0) is taken as the landing position. The foot takes the orientation of the surface on which it lands.

A.2. Transition with turning ($\alpha \neq 0$)

Turning motion is planned to ensure that the gait remains repeatable, i.e., after turning by a certain angle α, the biped should be in the same configuration in its local frame as it was prior to the turning operation. Furthermore, the turning procedure is designed in such a way that after $2\pi/\alpha$ rotations, the biped will return to the same state. Figure 23(b) shows the view of the biped normal to the supporting surface. The initial state is defined by the coordinate frames attached at P_i and Q. It is assumed that the two reference points lie on two concentric circles with radii r_o and r_i. The initial distance between the two feet along the direction of the movement is sl_1. At the time of turning, it is *constrained* that the total movement by the swing be $2sl_1$, i.e., $|P_1P_2| = 2sl_1$. In other words, no change in step length is allowed at the time of turning. Repeated motions of step length $2sl_1$ inscribe a regular polygon inside the outer circle. This is utilized to give the radius of the circle; a mathematical expression is provided by the following relation:

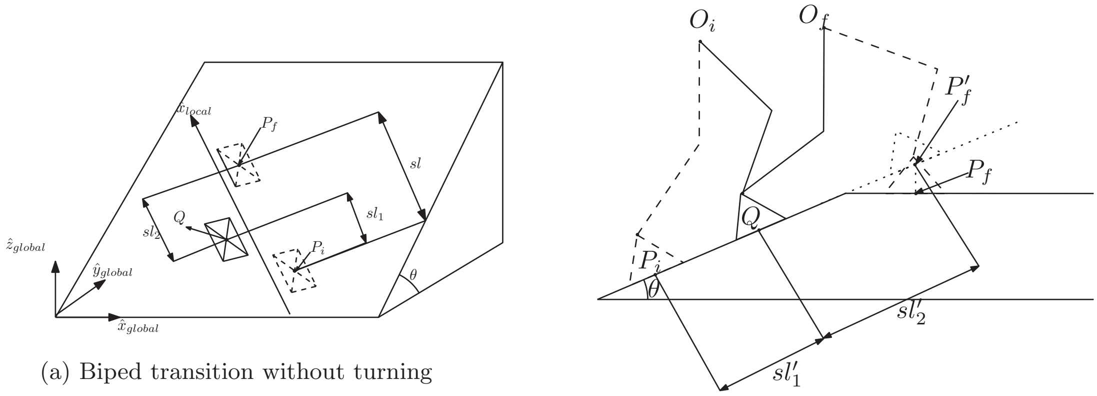
$$r_o = \frac{sl_1}{\sin(\alpha/2)} \tag{A1}$$

Furthermore, the radius of the inner circle (r_i) is defined to be

$$r = r_o - w_h \tag{A2}$$

where w_h is the width of the pelvis. Once the radii of the two concentric circles are known, the current position of the feet and information regarding sl_1 and α suffice in obtaining the center of the circle and the next step position. Figure 23(a) shows the generated motion for a turning angle $\alpha = 30°$ and $sl_1 = 0.075$ m. The turning was designed carefully using ideas of planar geometry, and this restricts the use of turning while transitioning from one slope to other.

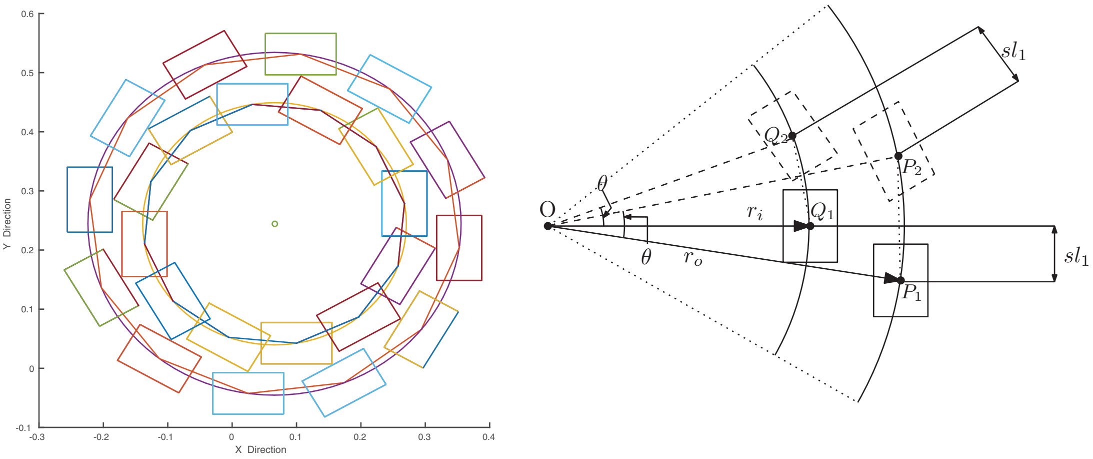
https://doi.org/10.1017/S0263574718000188 Published online by Cambridge University Press



(a) Biped transition without turning

(b) Biped transition: Slope change

Fig. 22. Biped transitions over slopes. (a) Biped transition without turning. (b) Biped transition: Slope change.



(a) Normal View of Biped Turning

(b) Turning methodology

Fig. 23. Turning motion of biped. (a) Normal view of biped turning. (b) Turning methodology.

https://doi.org/10.1017/S0263574718000188 Published online by Cambridge University Press