Generalizations of the Capture Point to Nonlinear Center of Mass Paths and Uneven Terrain

Oscar E. Ramos and Kris Hauser

***Abstract*— The classical Capture Point (CP) technique allows biped robots to take protective footsteps in case of a push or other disturbance, but only applies to flat terrain and a horizontally-moving Center of Mass (CoM). This paper generalizes the Capture Point technique to arbitrary terrains and CoM paths. Removing the CoM path constraint leads to an infinite number of Capture Points, each corresponding to a different path. A numerical algorithm is presented that, given an uneven terrain, enumerates Capture Points and their respective CoM trajectories. It is suitable for real-time usage as it produces all capture points in less than a millisecond on a standard PC. Proof-of-concept results are also demonstrated on a humanoid robot in a dynamic simulator.**

I. Introduction

Maintaining balance in the presence of disturbances is one of the main challenges in humanoid robotics. Small disturbances can be compensated through active feedback control of joint torques while maintaining the same stance leg, based on simplified models that consider linear momentum [1], [2] or angular momentum [3], [4]. However, larger disturbances, such as a strong push, require a reactive step. As a result, methods to choose foot placements for reactive steps has become a topic of recent interest.

Several approaches have been presented to address this problem. For example, one approach uses Model Predictive Control (MPC) to generate walking patterns or to determine adaptive foot positioning under several constraints including Center of Mass (CoM) objectives [5], [6]. It has been also suggested that using MPC, several steps can be planned ahead to recover balance [7], [8]. Proper ground reaction forces can also be enforced by locally modifying the planned foot landing position [9], and disturbances can be absorbed by modifying future steps, combining disturbance suppression with reactive stepping [10]. Another popular approach to determine a stepping point is the so-called Capture Point (CP) [11], which has been used for push recovery [12], walking pattern generation [13], and reaching of far objects via the integration in a whole body control framework [14].

However, these existing approaches make several simplifying assumptions for computational reasons that limit their applicability to general terrains. First, they select a footstep locomotion assuming the ground is perfectly flat. It may still be possible to apply them to flat terrain with small irregularities by applying local modifications to adapt the foot orientation to the terrain. But this approach is unsuitable for very uneven terrain, which often requires a change in footstep to adapt to large irregularities. Moreover, existing approaches usually assume that a fixed path for the CoM (e.g., a horizontal line) is given, which ignores the capability of the robot's legs to extend and contract in order to adapt to changes of terrain height, and to help reduce momentum and absorb impact. A recent work, [15], addresses the latter problem using capture points but it does not give a method for planning footstep positions.

Oscar E. Ramos and Kris Hauser are with the Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708, USA {oscar.ramos, kris.hauser}@duke.edu

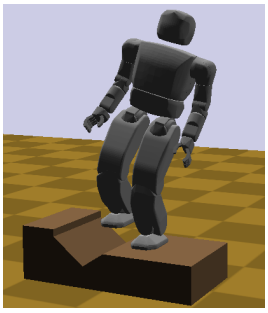


Fig. 1: Example of an uneven terrain where the robot must carefully choose a reactive footstep if it is laterally pushed.

This paper proposes a new methodology to generalize the Capture Point for balance recovery in two ways: first, CoM trajectories are allowed to describe more general curves such as a line with a slope and a parabola; second, the landing terrain is not constrained to a horizontal surface allowing us to handle arbitrary uneven terrain such as the one shown in Fig. 1. To achieve this, this paper presents a 2D nonlinear inverted pendulum model which extends the standard linear inverted pendulum (LIP) [16] model to the case of nonlinear trajectories. Whereas in the linear case the computation of the CP is analytical, in the general case there is no closed form solution, and we resort to a numerical approach. We use a direct shooting method to solve the differential equations, and we present methods for (i) solving for a Capture Point given a CoM trajectory, (ii) solving for a CoM trajectory that makes a footstep location a Capture Point, and (iii) calculating a discretized set of CPs, and a CoM trajectory for each CP, on a given terrain. A single degree of freedom in the CoM path is sufficient, and in this paper we use the family of quadratic curves.

The proposed algorithm is fast and can produce a set of CPs for an arbitrary polygonal 2D terrain in less than a millisecond on a standard PC. This paper presents numerical examples of the algorithm, as well as a proof-of-concept demonstration on a simulated Hubo-II+ robot off balance, using our algorithm to find a Capture Point on uneven terrain.

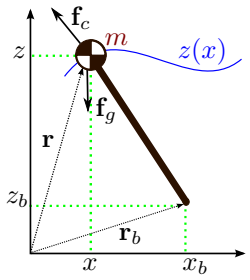


Fig. 2: Nonlinear Inverted Pendulum (NIP) in 2D, with fixed base and motion constrained to the curve $z(x)$.

II. Inverted Pendulum and Capture Point Models

In the standard linear inverted pendulum (LIP) model [16], the motion of the Center of Mass (CoM) is assumed to be purely horizontal. Combined with an assumption of flat ground, the Capture Point [11] can be determined in closed form. However, this model unnecessarily constrains the height of the CoM, which restricts the robot from using vertical displacement to conform to the terrain, to dissipate momentum, or to maintain rigidly extended legs during walking.

This section presents the *Nonlinear Inverted Pendulum* (NIP), a generalization of the LIP model that allows the CoM to move about an arbitrary nonlinear curve. For background it also reviews the LIP model, which specializes NIP to linear (non-horizontal) CoM motion; and the analytical definition of a Capture Point for the LIP model. In future sections we will present methods for calculating the generalized Capture Point.

A. Inverted Pendulum Model

The inverted pendulum is a simplified 2D model that is widely used to approximate the dynamics of a biped robot. It only considers the motion of its CoM in a plane in response to a single point of contact, leading to a 2D state space that is more mathematically and computationally tractable than the full dynamics of the robot.

The model describes the dynamics of the robot under the following assumptions.

- The total mass of the robot is represented as a point mass m located at its Center of Mass (CoM).
- The robot legs are massless, attach at the CoM, and can extend arbitrarily.
- One foot of the robot makes point contact with the ground (this point forms the pendulum base).
- A force may be applied directly along the axis of the leg (which forms the pendulum rod). No perpendicular forces or torques may be applied at the contact.
- Friction is sufficiently high to keep the point contact from slipping.
- The motion of the robot CoM is constrained to lie along a curve $z(x)$.

Let the CoM position be $\mathbf{r} = (x, z)$ and the pendulum base position be $\mathbf{r}_b = (x_b, z_b)$ with respect to some reference frame, as Fig. 2 shows. Let $\mathbf{f}_g = (0, -mg)$ be the gravity force, where g is the acceleration of gravity, and $\mathbf{f}_c$ be the force acting on the CoM due to the contact reaction. Since there is a single contact point, $\mathbf{f}_c$ is oriented in the direction of the massless rod and can be represented as

$$\mathbf{f}_c = (\mathbf{r} - \mathbf{r_b})f$$

where

$$f = \frac{\|\mathbf{f}_c\|}{\|\mathbf{r} - \mathbf{r_b}\|}$$

is a scalar that depends on the magnitude of the contact force and the CoM position. The total force acting on the CoM is the sum of both the gravity and contact forces $\mathbf{f}_{tot} = \mathbf{f}_c + \mathbf{f}_g$ and its components are given by

$$\mathbf{f}_{tot} = \begin{bmatrix} (x - x_b)f \\ (z - z_b)f - mg \end{bmatrix}. \tag{1}$$

Provided that the CoM motion lies along the 2D curve described by $z(x)$, as depicted in Fig. 2, its vertical velocity is $\dot{z} = \frac{dz}{dx}\dot{x}$, and its acceleration is given by

$$\ddot{\mathbf{r}} = \begin{bmatrix} \ddot{x} \\ \frac{dz}{dx}\ddot{x} + \frac{d^2z}{dx^2}\dot{x}^2 \end{bmatrix}. \tag{2}$$

Replacing (1) and (2) in the dynamic equation of motion $\mathbf{f_{tot}} = m\ddot{\mathbf{r}}$, it can be shown that the horizontal dynamics of the 2D nonlinear inverted pendulum is:

$$\ddot{x} = \frac{(x - x_b)\left(g + \dot{x}^2 \ \frac{d^2z}{dx^2}\right)}{(z - z_b) - (x - x_b)\frac{dz}{dx}} \tag{3}$$

and its vertical dynamics is:

$$\ddot{z} = \frac{(z - z_b)\dot{x}^2 \ \frac{d^2z}{dx^2} + (x - x_b)g\frac{dz}{dx}}{(z - z_b) - (x - x_b)\frac{dz}{dx}} \tag{4}$$

with $(z - z_b) - (x - x_b)\frac{dz}{dx} \neq 0$ in both (3) and (4).

The model is ill-defined (i.e., does not yield a unique solution) when the tangent to the curve at point (x, z) coincides with the line from the pendulum base to that point. Specifically, this occurs when $\frac{dz}{dx} = \frac{z - z_b}{x - x_b}$, or both $x = x_b$ and $z = z_b$. Let these tangency points be (x_{t_i}, y_{t_i}) as Fig. 3 shows. A physical intuition is that at this instant the velocity of the CoM is parallel to the pendulum rod, and the magnitude of the contact force dictates the magnitude of acceleration $(\ddot{x}, \ddot{z})$. To resolve this ambiguity, we might either assume the contact force or the acceleration to be zero. However, we will later show that this case can be safely ignored for a certain class of CoM paths.

B. Motion Constrained to a Line

In the case of linear CoM motion it is possible to solve the second-order differential equations (3), (4) analytically. Although the LIP, as typically presented, usually considers only horizontal CoM motion, this section derives a closed-form solution that also applies when the slope is nonzero.

Given the initial conditions (x_0, z_0) for the position of the CoM, and $(\dot{x}_0, \dot{z}_0)$ for its velocity, the line constraining the motion is represented as $z(x) = k(x - x_0) + z_0$, where

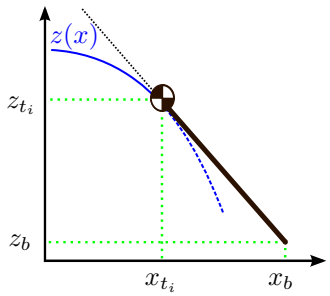


Fig. 3: At (x_{t_i}, z_{t_i}), the pendulum trajectory $z(x)$ becomes tangent to the pendulum rod generating an undefined dynamic behavior. The blue line depicts the trajectory, and the dashed blue line, the trajectory it would have if the motion continued.

$k = \frac{\dot{z}_0}{\dot{x}_0}$ is the slope. Setting $\frac{dz}{dx} = k$ and $\frac{d^2z}{dx^2} = 0$ in (3) and (4), one obtains horizontal dynamics

$$\ddot{x} = \frac{(x - x_b)\ g}{k(x_b - x_0) + z_0 - z_b} \tag{5}$$

and vertical dynamics

$$\ddot{z} = \frac{(x - x_b)\ g\ k}{k(x_b - x_0) + z_0 - z_b}. \tag{6}$$

It is observed that $\ddot{z} = \ddot{x}\ k$, which also follows from the second time derivative of the line equation. The horizontal motion (5) can be rewritten as

$$\ddot{x} - wx = -wx_b \tag{7}$$

where

$$w = \frac{g}{k(x_b - x_0) + z_0 - z_b}$$

Using the line equation, the denominator of w can be written as $z(x_b) - z_b$, i.e., the CoM height when it is above (or below) the pendulum base. Assuming $z(x_b) > z_b$, (i.e., $w > 0$) the solution to (7) is

$$\begin{aligned} x(t) = &\frac{1}{2}e^{\sqrt{w}t}\left(x_0 - x_b + \frac{\dot{x}_0}{\sqrt{w}}\right) \\ &- \frac{1}{2}e^{-\sqrt{w}t}\left(x_b - x_0 + \frac{\dot{x}_0}{\sqrt{w}}\right) + x_b \end{aligned} \tag{8}$$

with its first time derivative equal to

$$\begin{aligned} \dot{x}(t) = &\frac{1}{2}e^{\sqrt{w}t}\left(\sqrt{w}(x_0 - x_b) + \dot{x}_0\right) \\ &+ \frac{1}{2}e^{-\sqrt{w}t}\left(\sqrt{w}(x_b - x_0) + \dot{x}_0\right). \end{aligned} \tag{9}$$

The vertical trajectory can be found integrating the vertical dynamics or, more easily, using the line equation directly. When $z(x_b) < z_b$ (i.e., $w < 0$), the solution to (7) is oscillatory.

C. Definition of Capture Point

The *Capture Point* (CP), introduced in [11] and also referred to as the *Extrapolated Center of Mass* [17], is the point on the ground $\boldsymbol{\xi} = (\xi_x, \xi_z)$ where the robot should step to in order to come to a complete rest, via some prescribed motion model.

In the case of a nonlinear inverted pendulum model, inspection of (3) and (4) shows that a stationary point of this system can only be reached when the system evolves towards $\dot{x} = 0$ and $x = x_b$ as time approaches infinity:

$$\lim_{t\to\infty} x(t) = x_b \quad \text{and} \quad \lim_{t\to\infty} \dot{x}(t) = 0. \tag{10}$$

That is, the pendulum velocity at infinity must be null and the horizontal pendulum trajectory must converge to x_b (the pendulum stops above its base). In this case, the Capture Point is given by the pendulum base: $(\xi_x, \xi_z) = (x_b, z_b)$. It is important to note that this limit does not hold for any arbitrary (x_b, z_b).

D. Analytic Capture Point for LIP on Flat Ground

The classic Capture Point can be analytically solved given a linear CoM motion and a horizontal terrain (i.e., the value of z_b is fixed). In the standard case where the final CoM lies above z_b, the horizontal velocity limit in (10) holds when the coefficient of the unbounded term $e^{\sqrt{w}t}$ in (9) becomes zero; that is, $\sqrt{w}(x_0 - x_b) + \dot{x}_0 = 0$. Replacing the value of w, this condition becomes

$$(x_0 - x_b)\sqrt{\frac{g}{k(x_b - x_0) + z_0 - z_b}} + \dot{x}_0 = 0$$

and solving for x_b, this leads to

$$\xi_x = x_0 + k\frac{\dot{x}_0^2}{2g} + \dot{x}_0\sqrt{\frac{z_0 - \xi_z}{g} + \left(\frac{k\dot{x}_0}{2g}\right)^2} \tag{11}$$

which is the horizontal coordinate of the Capture Point with fixed height $\xi_z = z_b$.

We note that when $w < 0$, the pendulum dynamics is oscillatory and no point can satisfy the conditions in (10) leading to a Capture Point, unless a damping factor is added. In this case, the final CoM would be below the pendulum base (over-hanging pendulum) and any base point would be a suitable Capture Point. It is unrealistic to use such solutions using typical biped feet, but hypothetically such solutions may be interesting if the robot were permitted to use hands (such as the overhead straps used by standing bus passengers).

The Capture Point is typically described with the CoM following a horizontal trajectory ($k = 0$). But we observe that the CP may be changed by choosing alternate slopes, or terrains of alternate height. Fig. 4 depicts the forward travel distance of the CP, ξ_x for different slopes k and landing terrain heights ξ_z. As the slope decreases or the terrain height increases, the forward travel distance becomes shorter. It has also been shown that the Capture Point changes significantly if the pendulum length is assumed to be constant [18].

Capture Point variability is important to consider, because the CP must be intercepted using the swing leg, and shorter travel distances may lead to faster balance recovery. Moreover, far capture points may simply be unreachable. The problem, however, is that all but one line of varying slope k do not match the initial conditions of the CoM velocity $k = \dot{z}_0/\dot{x}_0$. As a result, to exploit CP variability, the CoM will need to travel on a nonlinear curve.

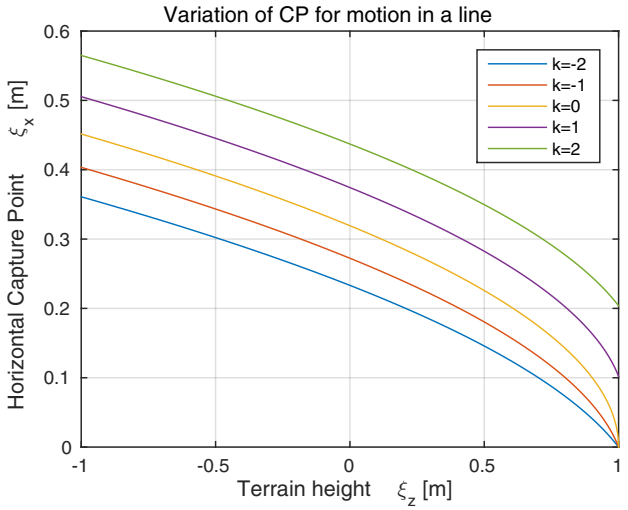


Fig. 4: Horizontal distances of Capture Points with varying terrain heights and CoM paths slopes k. The example shown uses $x_0 = 0$, $\dot{x}_0 = 1.0$, $z_0 = 1.0$. The yellow line ($k = 0$) is the CP under horizontal motion, as presented in [11]. (This figure best viewed in color)

III. Proposed Algorithms

To allow a CoM to move on a nonlinear path, we resort to numerical methods. We found that a shooting and bisection method is reliable and can calculate solutions in a few iterations, typically without getting stuck in local minima. Moreover, the method can find Capture Points on a terrain modeled as a non-horizontal line segment, which cannot be achieved with the classical Capture Point formulation. The chosen trajectory is quadratic since it is the least polynomial degree that allows for one extra degree of freedom when respecting the initial velocities.

Based on the shooting method, we present three algorithms for working with capture points under a 2D NIP model.

1) Calculate a Capture Point (ξ_x, ξ_z) given a CoM path $z(x)$, up to a desired tolerance.
2) Calculate parameters of a parabolic CoM path $z(x) = ax^2 + bx + c$ that make a footstep location (x_b, z_b) a capture point (i.e., $(\xi_x, \xi_z) = (x_b, z_b)$) up to a desired tolerance.
3) Enumerate all candidate Capture Points and their parabolic CoM paths for the terrain, given some discretization resolution.

The common input to each method are the CoM initial conditions (x_0, z_0), $(\dot{x}_0, \dot{z}_0)$, and a polygonal representation of the terrain. The algorithms are also given bounds on final height of the CoM above the terrain:

$$z_{min} \leq z(x_b) \leq z_{max} \tag{12}$$

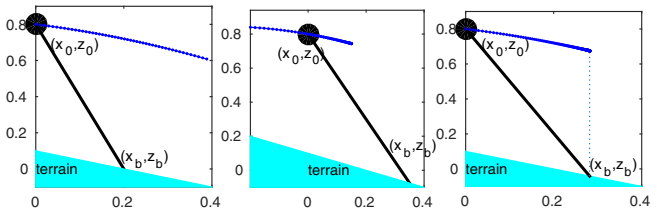
These constraints give an approximation to kinematic reachability.

A. Capture Point Computation by Shooting and Bisection

Without loss of generality, it can be assumed that $x_b > x_0$ and $\dot{x} > 0$ since, if $x_b < x_0$, there is no way the pendulum can stop, and if $\dot{x} < 0$ a simple inversion of the x axis can be done. A procedure to find a Capture Point that satisfies the conditions (10) in finite time is based upon the following observation. Given the initial conditions and curve parameters, the following three cases arise.

1) *Overshooting:* if the horizontal component of the pendulum base x_b is too close to the initial CoM horizontal position, the pendulum will overshoot it. In this case $x_f > x_b$, where x_f is the final x.
2) *Undershooting:* if x_b is too far, the CoM will not be able to reach it and it will begin to move backwards. In this case, $\dot{x}_f < 0$.
3) *Capture Point:* if x_b is located at the Capture Point, the CoM will stop above it.

An illustration of these cases is depicted in Fig. 5.



(a) Overshoot (b) Undershoot (c) Capture Point

Fig. 5: Idea of the numerical approach. The initial position of the CoM is at $(x, z) = (0,\ 0.8)$. In (a), the pendulum base is too close so the pendulum will overshoot the CP. In (b), the base is too far, the pendulum does not have enough energy to reach the CP and will start moving backwards. In (c), the CoM stops above its base, which becomes a CP.

The basic idea of bisection is to find an interval of points on the terrain in which closer points overshoot and farther points undershoot, and then to repeatedly bisect that interval. This repeats until a capture point is obtained within some tolerance. Namely, we use the tolerance:

$$|x_b - x_n| < \delta_p \quad \text{and} \quad |\dot{x}_n| < \delta_v \tag{13}$$

where δ_p and δ_v are arbitrarily small tolerances and $n \in 2, \cdots, N$ is a time less or equal to the maximum allowable time (if the condition is satisfied at $n < N$, it will also be true at k such that $n < k \leq N$). A maximum number of steps i_{max} is also defined but the algorithm finds a solution typically in around 10 to 15 steps.

Since the terrain is modeled as a set of line segments, we can initialize intervals of candidate capture points to the extrema of each segment. Let the line segment describing the terrain have extremes $P_1 = (x_1, z_1)$ and $P_2 = (x_2, z_2)$, such that $x_0 < x_1 < x_2$ as Fig. 6 shows. Hence, the search interval is initialized to $x_1 < x_b < x_2$. Later we will present methods for pruning the search interval even further.

The algorithm is shown in Alg. 1. It also requires a maximum integration time parameter, specified by the number of discrete samples N as well as the state $\mathbf{x}_0 = [x_0\ \dot{x}_0\ z_0\ \dot{z}_0]^T$. The current search range is $[p_0, p_f]$ and is bisected at each outer loop until the tolerance is reached, or up to N_{it} steps have been reached. The function `dynamics` provides an integration step of the NIP dynamics. If at some point the

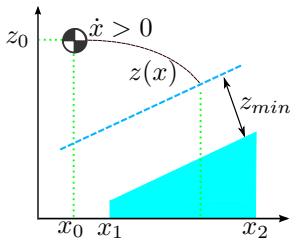


Fig. 6: By choosing x_b between x_1 and x_2 (or a reduced subset), and updating z_b according to the terrain, a CP can be found. The minimum allowable height of the pendulum CoM with respect to the terrain is z_{min}.

pendulum is overshooting ($x_n > x_b$), it will continue overshooting; and if at some point it is undershooting ($\dot{x}_n < 0$), it will continue undershooting. In either case, the integration can be halted early. The value of x_b is updated using a bisection of the interval $[p_0, p_f]$, and then the value of z_b is updated using the terrain equation.

Since we integrate the dynamics numerically, the precision of the final results are constrained by the sampling time dt as well as the tolerances δ_p, δ_v. The algorithm is resolution-complete in the sense that it will find a CP if it exists for the given CoM trajectory. If a result is returned, the result also respects the CoM kinematic limits.

Algorithm 1 Compute CP for a CoM path and a terrain

Input: $z(x)$, $\mathbf{x}_0$, P_1, P_2, N
Output: (ξ_x, ξ_z)

$p_0 \leftarrow x_1$, $p_f = x_2$
$x_b \leftarrow p_f$
for $i \leftarrow 1, N_{it}$ **do**
  $z_b \leftarrow z(x_b)$
  for $n \leftarrow 1, N$ **do**
    $x_n, \dot{x}_n \leftarrow$ dynamics$(x_{n-1}, z_{n-1}, x_b, z_b, z, g, dt)$
    if ($|x_b - x_n| < \delta_p$ and $|\dot{x}_f| < \delta_v$) (tolerance reached) **then**
      return (x_b, z_b), "found"
    else if $x_n > x_b$ (overshoot) **then**
      if $i = 1$ **then**
        return (x_b, z_b), "not found"
      end if
      $p_0 \leftarrow x_b$
      break
    else if $\dot{x}_n < 0$ (undershoot) **then**
      $p_f \leftarrow x_b$
      break
    end if
  end for
  $x_b \leftarrow \frac{p_0 + p_f}{2}$
end for
return (x_b, z_b), "not found"

B. Initializing the Bisection Interval

To initialize the bisection procedure with as small an interval as possible, we identify a region for potentially valid Capture Points. For a curve $z(x)$ constraining the trajectory of the pendulum, let (x_{t_i}, z_{t_i}) be points of tangency along the curve that intersect the pendulum base (x_b, z_b) and therefore satisfy the condition

$$\frac{dz}{dx}\Big|_{x=x_{t_i}} = \frac{z_{t_i} - z_b}{x_{t_i} - x_b}. \tag{14}$$

These points generate an undefined dynamic behavior in (3) and (4). Given the pendulum base (x_b, z_b), and the initial CoM position (x_0, z_0), let $\mathcal{F}(x_b, z_b) \subset \mathbb{R}^2$ be a set with components $(x(t), \dot{x}(t))$ such that, $\forall t > t_0$:

1) $\dot{x}(t) \geq 0$
2) $x(t) \in [x_0, x_b]$
3) $x_{t_i} \notin [x_0, x_b]$, $\forall x_{t_i}$, if they exist.

Starting with the initial conditions (x_0, z_0), $(\dot{x}_0, \dot{z}_0)$ at $t = t_0$, with $\dot{x}_0 > 0$, the point (x_b, z_b) is a candidate for a Capture Point if $(x, \dot{x}) \in \mathcal{F}(x_b, z_b)$. It will be a Capture Point if (10) is satisfied in finite time.

C. Making a Given Point a Capture Point via Parabola Search

A given point (x_b, z_b) on a landing terrain can be turned into a Capture Point if the proper path for the pendulum CoM is chosen. We use a quadratic path with variable curvature (defined by a) in an attempt to obtain a Capture Point. Like before, we use a bisection approach, but the search is along the value of a. As values of a decrease, the parabola curves downwards, and the pendulum gains energy overshooting the desired x_b. As a increases, the trajectory curves upwards, the pendulum loses energy and eventually begins to move backwards.

Let the curve constraining the motion of the inverted pendulum be given by the following parabola

$$z = a(x - x_0)^2 + b(x - x_0) + z_0. \tag{15}$$

The velocity of (15) is $\dot{z} = (2a(x - x_0) + b)\dot{x}$ which constrains the value of b given the initial conditions. That is, $b = \frac{\dot{z}_0}{\dot{x}_0}$ is a parameter completely determined by the initial velocity. Hence, the curvature a is the only free parameter of the parabola.

During the search, special care should be taken to deal with the ill-defined situation where the denominator is zero, since it would lead to erroneous results. A straightforward geometric analysis using (14) and (15) shows that for the quadratic curve there exist two points of tangency which occur at

$$x_{t_i} = x_b \pm \sqrt{(x_b - x_0)^2 + \frac{z_0 - z_b + b(x_b - x_0)}{a}} \tag{16}$$

Let these points be ordered as $x_{t_1} < x_{t_2}$. When the trajectory defined by the dynamics of the quadratic system reaches one of those points, the dynamic equations become undefined. Therefore, the forward shooting should not continue after these values. The value of the parameter a that makes the square root in (16) equal to zero is $a_{cr} = \frac{z_b - z_0 - b(x_b - x_0)}{(x_b - x_0)^2}$ where $b_{cr} = \frac{z_b - z_0}{x_b - x_0}$ and can be obtained using basic geometry. Using a simple geometric analysis of the quadratic

curve, it can be shown that the quadratic dynamics behavior is defined beforehand in the following cases.

- If $a = a_{cr}$, the dynamics is defined while $x(t) < x_b$.
- If both $b > b_{cr}$ and $a < a_{cr}$, or both $b < b_{cr}$ and $a > a_{cr}$, the dynamics is defined while $x(t) < x_{t_1}$.
- If both $b > b_{cr}$ and $a > 0$, or both $b < b_{cr}$ and $a < 0$, the dynamics is defined while $x(t) \in [x_{t_1}, x_{t_2}]$.

In all other cases, the quadratic dynamics is well defined in all the points belonging to the trajectory $x(t) \in \mathbb{R}$.

To further reduce the search, a minimum and maximum values for the curvature, a_{min} and a_{max}, can be readily found from (15) (using some initial position x_0, z_0) based upon the minimum and maximum height, z_{min} and z_{max}, that the CoM can have above the landing point (x_b, z_b). These limits above the terrain are schematically shown in Fig. 7 and are determined by the kinematic constraints of the robot. In case no a in that range exists, then there exists no feasible CP given the kinematic constraints.

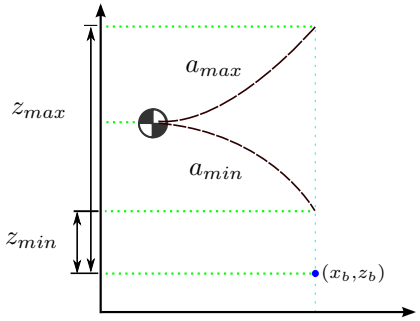


Fig. 7: By choosing the proper curvature for the parabola, the point (x_b, z_b) can be turned into a CP. The search for the curvature value (a) will be in the range $[a_{min}, a_{max}]$, which is determined by the allowable heights z_{min} and z_{max}.

Alg. 2 lists the algorithm to find the parabolic curve (15) and is very similar to the algorithm used in the previous case, except the desired footstep location (x_b, z_b) is given rather than the path $z(x)$. The value of a is updated using a bisection between the maximum and minimum values. The condition for ending the iterations is the one in (13). Since the initial value of a is the maximum allowable, if it leads to an overshoot, there is no way to increase it more (to find a Capture Point), and no feasible CP exists. However, if a CP exists, the algorithm will find it.

D. Enumerating All Capture Points and CoM Paths

Given a terrain modeled by a line segment, described by its extremes (x_1, z_1) and (x_2, z_2), it is possible to find all Capture Points on it corresponding to parabolic paths, as a generalization of the previous algorithm. In this case, the curvature of the parabola (a) can be geometrically bounded using the limits of the landing terrain, as Fig. 8 depicts.

The steps to find CPs and their respective CoM trajectories in this case can be summarized as:

1) Find the curves (described by the value of a) to make the segment extremes x_1, x_2 become Capture Points using Alg. 2. If successful, proceed to step 2). Otherwise:

Algorithm 2 Making a point become a Capture Point

Input: (x_b, z_b), $\mathbf{x}_0$, N
Output: a

Compute $b = \frac{\dot{z}_0}{\dot{x}_0}$, a_{min}, a_{max}
$a \leftarrow a_{max}$
for $i \leftarrow 1, N_{it}$ **do**
  for $n \leftarrow 1, N$ **do**
    $x_n, \dot{x}_n \leftarrow$ dynamics$(x_{n-1}, z_{n-1}, x_b, z_b, a, b, g, dt)$
    if ($|x_b - x_n| < \delta_p$ and $|\dot{x}_f| < \delta_v$) **then**
      return a, "found"
    else if $x_n > x_b$ **then**
      if $i = 1$ **then**
        return a, "not found"
      end if
      $a_{min} \leftarrow a$
      break
    else if $\dot{x}_n < 0$ **then**
      $a_{max} \leftarrow a$
      break
    end if
  end for
  $a \leftarrow \frac{a_{min}+a_{max}}{2}$
end for
return a, "not found"

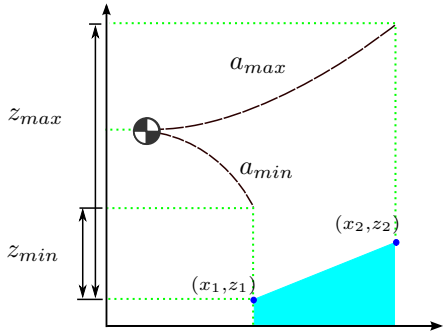


Fig. 8: Given a terrain, the curves (defined by a) that will generate Capture Points along the terrain are searched.

- If no feasible CP exists at x_1 due to a final CoM too high over x_1, then no feasible CP exists in the segment (since, as x_1 increases, a also increases).
- If no feasible CP exists at x_2 due to a final CoM too low over x_2, then no feasible CP exists in the segment (since, as x_2 decreases, a also decreases).
- In other cases, move the extremes (increase x_1 or decrease x_2 along the terrain) and repeat step 1).

2) Increase the lowest point by Δx and use Alg. 2 on this new point to find the curve.

By repeating the last step, the output of the approach are CPs along the terrain with their respective CoM trajectories. It has been empirically found that as x_b moves farther, the value of a needed to make it a Capture Point also increases. This observation is used as a heuristic to hasten the computation in step 2.

More complex terrains can be composed of a set of line

segments, treating each line segment separately and finding CPs on each segment. In this case, the CPs and the CoM trajectories computed at the boundary of the contiguous terrain segment can be used to hasten the search. For a long terrain the reachable leg space is also used to discard points that are far away from the initial CoM position.

IV. Results

The proposed algorithms for the computation of the Capture Point have been tested for several initial conditions and parabolic curves. An example, using initial conditions $(x_0, z_0) = (1.0,\ 0.8)$, $(\dot{x}_0, \dot{z}_0) = (1.0,\ -0.4)$ and a sampling time $dt = 0.04$ s, with a final time $t_f = 2$ s is shown in Fig. 9. In this case, an exhaustive computation of the Capture Points has been undertaken, for different terrain heights z_b ranging from -0.5m to 0.5m. The final pendulum CoM height is represented as z_f. Each point in the graph constitutes a Capture Point for the given terrain height and the given curve. Both graphs depict the same points but they are joined differently, based either on the same curve a or the same terrain height z_b. In these figures, the height limits of the CoM have been removed in order to illustrate the broad range of behaviors that could be obtained, but in practice these limits should be adjusted to match the kinematic limits of the robot. For different initial conditions, similar results are observed, with larger values of x_b generally obtained for larger initial velocities.

A simulation using a 2D inverted pendulum is shown in Fig. 10 which depicts an uneven terrain and the pendulum in its initial position. The computation time is around 0.1 ms (in C++) for the shown terrain, which can be considered real time for current humanoid robots' control loop. The approach in Sec. III-D computes the feasible CPs (shown in red) given some feasible final heights for the CoM to achieve (which would depend on the robot kinematics). Fig. 10 also shows blue points on the ground and their respective final CoM height to emphasize that, although CPs exist in those regions, they are not feasible.

To show the application in a robot, a simulation was done using the model of a humanoid robot and the dynamic simulator Klamp't (`http://klampt.org`). The robot starts in single support as Fig. 11 shows. Then an external force is applied at the chest of the robot from right to left (viewer's point of view). This force destabilizes the robot and it starts "falling". At this moment, a finite-horizon falling trajectory is estimated considering the time it would take the foot to get to the ground. This falling trajectory estimation, based on an inverted pendulum describing a circle, acts as a prediction to account for the required swing time. Then, several Capture Points are computed on the given ground using different curves, and the foot is set to move to the Capture Point closest to the robot. Although the model is 2D, the simulation is 3D since the constraining plane is taken to be in the direction of the force. For this problem the CoM was assumed to move only in the plane containing the lateral and vertical direction; in general, movement can be restricted to the plane through the CoM and perpendicular to both the

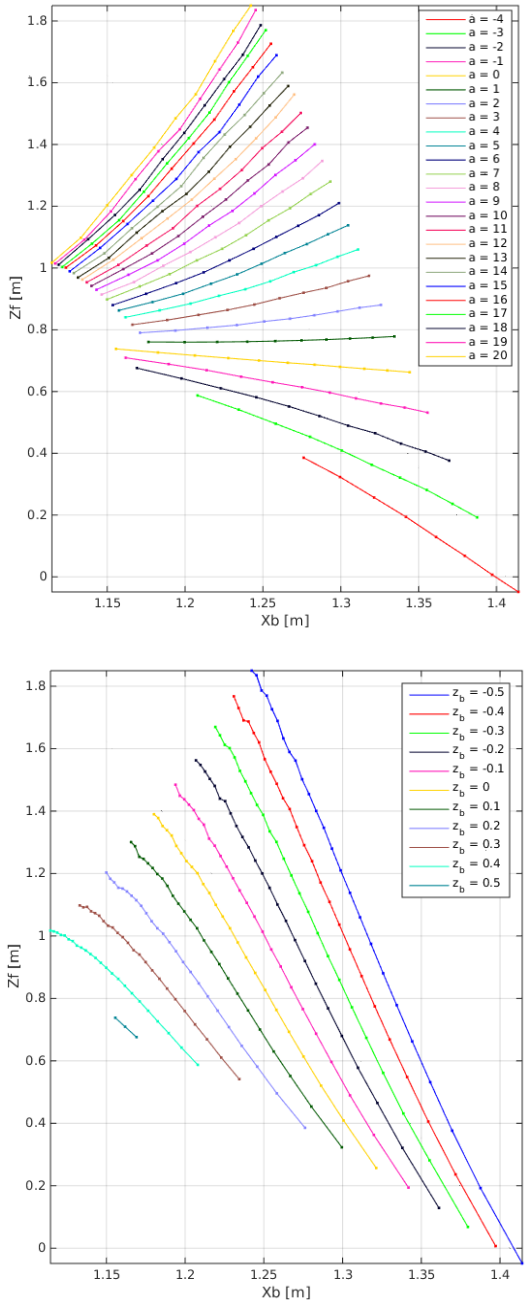


Fig. 9: Capture Points computed for values of a ranging from -4 to 20, and height of the terrain z_b from -0.5 to 0.5. (This figure best viewed in color)

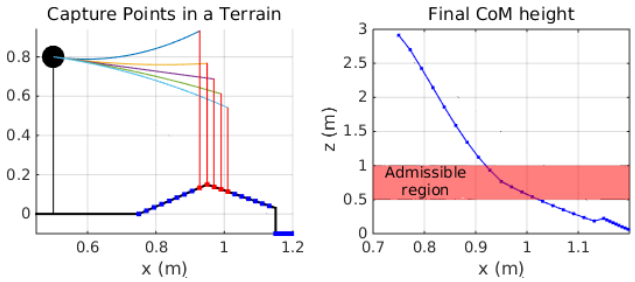


Fig. 10: Feasible Capture Points and some trajectories needed to achieve them given a terrain and certain initial conditions (left). The CoM height at the end of the trajectory is used as one of the criteria to determine the region that lies within the robot kinematic capabilities (right).

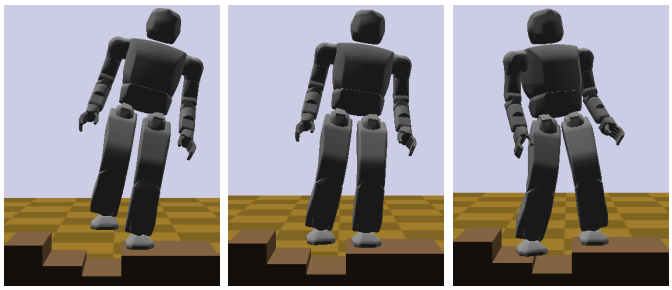


Fig. 11: Snapshots of the robot recovering after a lateral push by computing the Capture Points on an uneven terrain and stepping on one of them.

gravity vector and the CoM velocity. Also, a variation of the initial push magnitude determines different initial conditions and the computed point changes accordingly.

V. Conclusions

This paper has presented a nonlinear inverted pendulum (NIP) model, relaxing the horizontal CoM assumption used by the linear inverted pendulum model. Based on the NIP model, several Capture Points can be found depending on the CoM trajectory. Linear and quadratic trajectories have been considered, but it is possible to apply the proposed numerical approach to other types of trajectories. More importantly, given an uneven terrain, the proposed numerical approach is able to find Capture Points on the terrain, and the respective CoM trajectory, which cannot be achieved with previous approaches.

Future work will address the Capture Point using a nonlinear 3D model, as well as the effects of angular momentum on the computations. Capture Point based walking can be explored based on the proposed approach. We are also interested in implementing the current method on a real humanoid robot, and possibly using other limbs (e.g., hands) to aid in balance recovery.

References

[1] S. Kagami, F. Kanehiro, Y. Tamiya, M. Inaba, and H. Inoue, "Autobalancer: An online dynamic balance compensation scheme for humanoid robots," in *Fourth Int. Workshop on Algorithmic Foundations on Robotics*, 2000.

[2] S. Kudoh, T. Komura, and K. Ikeuchi, "The dynamic postural adjustment with the quadratic programming method," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2002.

[3] A. Macchietto, V. Zordan, and C. R. Shelton, "Momentum control for balance," in *ACM Transactions on Graphics (TOG)*, vol. 28, no. 3, 2009.

[4] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Resolved momentum control: Humanoid motion planning based on the linear and angular momentum," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2003.

[5] H. Diedam, D. Dimitrov, P.-B. Wieber, K. Mombaur, and M. Diehl, "Online walking gait generation with adaptive foot positioning through linear model predictive control," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2008, pp. 1121–1126.

[6] B. J. Stephens and C. G. Atkeson, "Push recovery by stepping for humanoid robots with force controlled joints," in *IEEE/RAS Int. Conf. on Humanoid Robots (Humanoids)*, 2010.

[7] Z. Aftab, T. Robert, and P.-B. Wieber, "Ankle, hip and stepping strategies for humanoid balance recovery with a single model predictive control scheme," in *IEEE/RAS Int. Conf. on Humanoid Robots (Humanoids)*, 2012.

[8] A. Sherikov, D. Dimitrov, and P.-B. Wieber, "Whole body motion controller with long-term balance constraints," in *IEEE/RAS Int. Conf. on Humanoid Robots (Humanoids)*, Madrid, Spain, 2014.

[9] T. Takenaka, T. Matsumoto, T. Yoshiike, T. Hasegawa, S. Shirokura, H. Kaneko, and A. Orita, "Real time motion generation and control for biped robot-4 th report: Integrated balance control," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2009.

[10] M. Morisawa, F. Kanehiro, K. Kaneko, N. Mansard, J. Sola, E. Yoshida, K. Yokoi, and J.-P. Laumond, "Combining suppression of the disturbance and reactive stepping for recovering balance," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2010.

[11] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: A step toward humanoid push recovery," in *IEEE/RAS Int. Conf. on Humanoid Robots (Humanoids)*, Genova, Italy, 2006.

[12] T. Koolen, T. de Boer, J. R. Rebula, A. Goswami, and J. E. Pratt, "Capturability-based analysis and control of legged locomotion," *The Int. Journal of Robotics Research (IJRR)*, vol. 31, no. 9, 2012.

[13] J. Englsberger, C. Ott, M. Roa, A. Albu-Schaffer, and G. Hirzinger, "Bipedal walking control based on capture point dynamics," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2011.

[14] O. Ramos, N. Mansard, and P. Souères, "Whole-body motion integrating the capture point in the operational space inverse dynamics control," in *IEEE/RAS Int. Conf. on Humanoid Robots (Humanoids)*, Madrid, Spain, 2014.

[15] J. Englsberger, T. Koolen, S. Bertrand, J. Pratt, C. Ott, and A. Albu-Schaffer, "Trajectory generation for continuous leg forces during double support and heel-to-toe shift based on divergent component of motion," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014.

[16] 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," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Maui, HI, USA, 2001.

[17] A. Hof, M. Gazendam, and W. Sinke, "The condition for dynamic stability," *Journal of biomechanics*, vol. 38, no. 1, pp. 1–8, 2005.

[18] Z. Li, C. Zhou, H. Dallali, N. G. Tsagarakis, and D. G. Caldwell, "Comparison study of two inverted pendulum models for balance recovery," in *IEEE-RAS International Conference on Humanoid Robots*, Madrid, Spain, 2014.