

This is the author's copy of the publication as archived with the DLR's electronic library at <http://elib.dlr.de>. Please consult the original publication for citation.

Realtime Limb Trajectory Optimization for Humanoid Running Through Centroidal Angular Momentum Dynamics

Sovukluk, Sait; Schuller, Robert; Engelsberger, Johannes; Ott, Christian

Copyright Notice

©2025 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Citation Notice

```
@INPROCEEDINGS{Sovukluk2025,  
  author={Sovukluk, Sait and Schuller, Robert and Engelsberger, Johannes and Ott, Christian},  
  booktitle={2025 IEEE International Conference on Robotics and Automation (ICRA)},  
  title={Realtime Limb Trajectory Optimization for Humanoid Running Through Centroidal Angular Momentum Dynamics},  
  year={2025},  
  volume={},  
  number={},  
  pages={404-410},  
  doi={10.1109/ICRA55743.2025.11127382}}
```

- [1] Sait Sovukluk, Robert Schuller, Johannes Engelsberger, and Christian Ott. Realtime limb trajectory optimization for humanoid running through centroidal angular momentum dynamics. In 2025 IEEE International Conference on Robotics and Automation (ICRA), pages 404–410, 2025. doi:10.1109/ICRA55743.2025.11127382.

Realtime Limb Trajectory Optimization for Humanoid Running Through Centroidal Angular Momentum Dynamics

Sait Sovukluk, Robert Schuller, Johannes Engelsberger, and Christian Ott

Abstract—One of the essential aspects of humanoid robot running is determining the limb-swinging trajectories. During the flight phases, where the ground reaction forces are not available for regulation, the limb swinging trajectories are significant for the stability of the next stance phase. Due to the conservation of angular momentum, improper leg and arm swinging results in highly tilted and unsustainable body configurations at the next stance phase landing. In such cases, the robotic system fails to maintain locomotion independent of the stability of the center of mass trajectories. This problem is more apparent for fast and high flight time trajectories. This paper proposes a real-time nonlinear limb trajectory optimization problem for humanoid running. The optimization problem is tested on two different humanoid robot models, and the generated trajectories are verified using a running algorithm for both robots in a simulation environment.

I. INTRODUCTION

Humanoid robot locomotion is mainly composed of two parts. The first part and the main objective of locomotion is translating the overall body mass (center of mass) to a desired point and at a desired rate. The second part is preserving a proper posture that is suitable to the nature of the selected locomotion method. The posture objective also splits into two parts: keeping the torso upright and swinging the feet in the direction of locomotion to prepare for the next step. This paper aims to determine how to swing the limbs so that the robotic system is ready for the next step while preserving a proper posture.

As the influence of the contact forces on the centroidal dynamics is significant [1], [2], motion generation for the center of mass (CoM) through a template or reduced models is very popular in the legged locomotion literature. These models are crafted to capture the ground reaction force patterns and centroidal dynamics well and are used to simplify some multi-body effects and limitations such that online control, planning, and decision-making are computationally feasible. The usage of centroidal or center of mass models in humanoid control appears in push recovery control through capture point [3]; walking control through linear inverted pendulum (LIP) [4], divergent component of motion

(DCM) [5], and spring-loaded inverted pendulum (SLIP) based walking [6], [7]; running control through SLIP [8], [9] and biologically inspired deadbeat control (BID) [10]. Control and planning through such models also appear in periodic jumping [11], bipedal backflip [12] and are also widely used in quadruped locomotion literature in single rigid body dynamics (SRBD) form [13]. The locomotion control literature is covered extensively in [14] and [15].

Even though the aforementioned methods showed success in simulation and hardware experiments, they do not capture the limb dynamics. All the methods assume the stance foot is placed at a specific location and do not include any information regarding the torso orientation, swing leg trajectories, and arm trajectories. Whole-body controllers handle this part of the locomotion control and require the user to select and tune these uncaptured trajectories. In the case of the slow pace and small step size walking, which covers the walking of almost all humanoid robots in the current market, the effect of limb swing is not that apparent mainly for two reasons. First, by the nature of walking behaviors, at least one foot is always in contact with the ground. Hence, the ground reaction forces can continuously be used for regulation as long as the friction cone allows. Second, the swing leg velocity is small due to the high step times and small foot displacements. Hence, the effect of leg swinging on the overall posture and dynamics is small. Such behaviors may not even require an arm swing behavior to regulate the angular momentum. As the walking gets more dynamic and the step length increases, the swing leg moves faster and affects on the overall dynamics more significantly. Such behaviors require more precise trajectory tuning both for the swing leg and arms. This problem is addressed in [16] through an optimization for the overall centroidal angular momentum target value.

In the case of running, which is the target of this paper, the aforementioned conditions get even more challenging. As the flight phase does not involve any contact with the ground, limb movement during this phase drastically affects the overall posture due to the conservation of angular momentum. Such behaviors require either very well-tuned limb swing trajectories or have to be limited with very short-lasting flight phases. This problem is more apparent while running through random environments where, due to lack of periodicity, each step may require different running trajectories along with their well-tuned limb swing characteristics. Unprecise tuning of limb trajectories results in ill-defined postural configurations and high body rotational velocities at the touchdown, which cannot be dissipated by the time and friction cone

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 819358).

Sait Sovukluk is with the Automation and Control Institute (ACIN), TU Wien, 1040 Vienna, Austria (e-mail: sovukluk@acin.tuwien.ac.at)

Robert Schuller and Johannes Engelsberger is with the Institute of Robotics and Mechatronics, German Aerospace Center (DLR), 82234 Weßling, Germany (e-mail: Robert.Schuller@dlr.de, johannes.engelsberger@dlr.de).

Christian Ott is with the Automation and Control Institute (ACIN), TU Wien, 1040 Vienna, Austria, and also with the Institute of Robotics and Mechatronics, German Aerospace Center (DLR), 82234 Weßling, Germany (e-mail: christian.ott@tuwien.ac.at).

limited stance phase control. Independent from the desired center of mass trajectory tracking performance, such systems inevitably fail to sustain their motions either immediately or as a result of the accumulated body orientation error. We demonstrate such behaviors in the supplemental video.

This paper focuses on online nonlinear constrained limb trajectory optimization through centroidal angular momentum dynamics for humanoid running. The objective of the optimization is to determine the liftoff configuration and flight phase joint trajectories such that at the next touchdown, the body orientation is minimal. The proposed optimization structure is independent of centroidal running trajectory generation methods. It requires only the desired stance leg position w.r.t. the CoM location at the moment of liftoff and touchdown. While doing so, we exploit some properties of the centroidal momentum matrix (CMM), reduce the size of the problem by some minimal simplifications, and construct a nonlinearly constrained optimization problem that can be solved in real-time. We implement our nonlinear optimization solver for the best efficiency and customization. Lastly, we verify our method in a simulation environment on two different robots. We also implement SLIP-based humanoid running trajectories and verify that the optimized flight phase joint trajectories result in minimal body orientation in the next touchdown phase. The running trajectories are constructed for long flight phases, even longer than the stance phase, such that they are more challenging and harder to control. Lastly, we share the computation time report and show that online implementation is possible.

II. SYSTEM DYNAMICS

Let \mathbf{q} be a set of configuration variables and $\boldsymbol{\nu} = (\boldsymbol{\nu}_b, \boldsymbol{\nu}_j)$ be the generalized velocity where $\boldsymbol{\nu}_b = (\mathbf{v}_b, \boldsymbol{\omega}_b) \in \mathbb{R}^6$ is the linear and angular velocity of the floating base and $\boldsymbol{\nu}_j \in \mathbb{R}^n$ is the generalized velocity of the joints. The well-known robotic system dynamics results in

$$\underbrace{\begin{bmatrix} \mathbf{M}_{bb} & \mathbf{M}_{bj} \\ \mathbf{M}_{jb} & \mathbf{M}_{jj} \end{bmatrix}}_{\mathbf{M}} \begin{bmatrix} \dot{\boldsymbol{\nu}}_b \\ \dot{\boldsymbol{\nu}}_j \end{bmatrix} + \mathbf{C}(\mathbf{q}, \boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{\tau}_g(\mathbf{q}) = \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\tau} \end{bmatrix} + \mathbf{J}_c(\mathbf{q})^\top \mathbf{f}_c, \quad (1)$$

where $\mathbf{f}_c \in \mathbb{R}^{3n_c}$ is the vector of contact forces for n_c number of contacts. The first six rows of (1) correspond to the floating base dynamics and are the underactuated portion of the system dynamics, i.e., the base dynamics cannot directly be driven by the instantaneous joint torques. Hence, the base movement is determined by the contact forces and inertial couplings.

Define $\mathbf{h}_G = [\mathbf{l}_G; \mathbf{k}_G] \in \mathbb{R}^6$ as the centroidal momentum, which is a combination of translational $\mathbf{l}_G \in \mathbb{R}^3$ and rotational (angular) $\mathbf{k}_G \in \mathbb{R}^3$ momentum. Then, based on [2] and [17], the first six rows of (1) are equivalent to

$$\dot{\mathbf{h}}_G = \begin{bmatrix} \dot{\mathbf{l}}_G \\ \dot{\mathbf{k}}_G \end{bmatrix} = \begin{bmatrix} m\mathbf{g} \\ \mathbf{0} \end{bmatrix} + \sum_{i=1}^{n_c} \begin{bmatrix} \mathbf{f}_{c,i} \\ (\mathbf{p}_{c,i} - \mathbf{p}_{\text{CoM}}) \times \mathbf{f}_{c,i} \end{bmatrix}, \quad (2)$$

where $\mathbf{g} \in \mathbb{R}^3$ is the gravitational acceleration vector, $\mathbf{f}_{c,i} \in \mathbb{R}^3$ is the contact force vector originating from $\mathbf{p}_{c,i} \in \mathbb{R}^3$,

$\mathbf{p}_{\text{CoM}} \in \mathbb{R}^3$ is the position of center-of-mass, and m is the total mass. The centroidal momentum dynamics (2) shows the significance of the contact forces on the body dynamics and is one of the main motivations of template model based locomotion approaches [1], [14]. The centroidal momentum is related to the generalized velocities through $\mathbf{h}_G = \mathbf{A}_G(\mathbf{q})\boldsymbol{\nu}$ where $\mathbf{A}_G \in \mathbb{R}^{6 \times (n+6)}$ is the centroidal momentum matrix (CMM) [1], [17].

III. PROPERTIES OF THE CENTROIDAL ANGULAR MOMENTUM MATRIX

In order to focus our discussion on the angular portion of the CMM, we decompose the linear and angular portions:

$$\mathbf{A}_G = \begin{bmatrix} \mathbf{A}_l \in \mathbb{R}^{3 \times (n+6)} \\ \mathbf{A}_k \in \mathbb{R}^{3 \times (n+6)} \end{bmatrix}. \quad (3)$$

The centroidal angular momentum matrix \mathbf{A}_k can also be decomposed into

$$\mathbf{A}_k = [\mathbf{A}_v \in \mathbb{R}^{3 \times 3} \quad \mathbf{A}_\omega \in \mathbb{R}^{3 \times 3} \quad \mathbf{A}_j \in \mathbb{R}^{3 \times n}], \quad (4)$$

where \mathbf{A}_v , \mathbf{A}_ω , and \mathbf{A}_j represent body translational, body rotational, and joint velocity portions, respectively.

Property 1 \mathbf{A}_v is a zero matrix.

Proof: Let ${}^G\mathbf{R}_B \in SO(3)$ be a rotation matrix from the body frame to the centroidal (CoM) frame. Additionally, let ${}^B\mathbf{X}_G^T \in \mathbb{R}^{6 \times 6}$ be momentum transformation from the floating base body frame to the centroidal frame [18]:

$${}^B\mathbf{X}_G^T = \begin{bmatrix} {}^G\mathbf{R}_B & \mathbf{0}_{3 \times 3} \\ {}^G\mathbf{R}_B \mathbf{S}({}^B\mathbf{p}_G)^\top & {}^G\mathbf{R}_B \end{bmatrix}. \quad (5)$$

The centroidal momentum matrix can be calculated as [17]

$$\mathbf{A}_G = \begin{bmatrix} \mathbf{A}_l \in \mathbb{R}^{3 \times (n+6)} \\ \mathbf{A}_k \in \mathbb{R}^{3 \times (n+6)} \end{bmatrix} = {}^B\mathbf{X}_G^T \begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} & \mathbf{M}_{13} \\ \mathbf{M}_{21} & \mathbf{M}_{22} & \mathbf{M}_{23} \end{bmatrix}, \quad (6)$$

where the rightmost matrix is the first six rows of the mass matrix: $\{\mathbf{M}_{11}, \mathbf{M}_{12}, \mathbf{M}_{21}, \mathbf{M}_{22}\} \in \mathbb{R}^{3 \times 3}$ and $\{\mathbf{M}_{13}, \mathbf{M}_{23}\} \in \mathbb{R}^{3 \times n}$. Similarly, decomposing angular portion of the centroidal momentum matrix as

$$\mathbf{A}_k = [\mathbf{A}_v \in \mathbb{R}^{3 \times 3} \quad \mathbf{A}_\omega \in \mathbb{R}^{3 \times 3} \quad \mathbf{A}_j \in \mathbb{R}^{3 \times n}]$$

results in

$$\mathbf{A}_v = {}^G\mathbf{R}_B \mathbf{S}({}^B\mathbf{p}_G)^\top \mathbf{M}_{11} + {}^G\mathbf{R}_B \mathbf{M}_{21}. \quad (7)$$

Similarly, based on [18] and [17]

$$\begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{21} & \mathbf{M}_{22} \end{bmatrix} = \begin{bmatrix} m_t \mathbf{1}_{3 \times 3} & m_t \mathbf{S}({}^B\mathbf{p}_G)^\top \\ m_t \mathbf{S}({}^B\mathbf{p}_G) & \bar{\mathbf{I}}_B \end{bmatrix}, \quad (8)$$

where $\bar{\mathbf{I}}_B(\mathbf{q}) \in \mathbb{R}^{3 \times 3}$ is the overall Cartesian inertia about the floating base body frame. Finally, the substitution of \mathbf{M}_{11} and \mathbf{M}_{21} from (8) into (7) yields

$$\mathbf{A}_v = m_t {}^G\mathbf{R}_B \mathbf{S}({}^B\mathbf{p}_G)^\top + m_t {}^G\mathbf{R}_B \mathbf{S}({}^B\mathbf{p}_G) = \mathbf{0}_{3 \times 3}. \quad (9)$$

The important implication of Property 1 is that the centroidal angular momentum dynamics is completely decoupled from

the translational body velocities. Knowing the initial condition, joint trajectories, and flight time, the body orientation evolution is the same for any set of translational body velocities. Hence, the resultant optimization problem does not require any translational trajectory information and is completely independent of running models.

Property 2 A_ω is always invertible.

Proof: Invertibility of A_ω is intuitively apparent and has a similar proof. Similar to (7)

$$A_\omega = {}^G R_B S({}^B p_G)^\top M_{12} + {}^G R_B M_{22}. \quad (10)$$

Substitution of M_{12} and M_{22} from (8) into (10) yields

$$A_\omega = m_t {}^G R_B S({}^B p_G)^\top S({}^B p_G)^\top + {}^G R_B \bar{I}_B. \quad (11)$$

where $\bar{I}_B = \bar{I}_{com} + m_t S({}^B p_G) S({}^B p_G)^\top$ [18] and \bar{I}_{com} represents the overall inertia with respect to the center of mass. Consequently

$$A_\omega = {}^G R_B \bar{I}_{com}. \quad (12)$$

Hence, A_ω is always invertible as it is a product of two invertible matrices. ■

The invertibility of A_ω implies that for a given centroidal angular momentum and joint velocities, there exists a finite body rotational velocity at any system configuration. As this work focuses on estimating the body's rotational velocity and minimizing its integration during the flight phase, Property 2 ensures that there exists a solution to the calculation for any joint configuration and velocity.

IV. FLIGHT PHASE BODY ORIENTATION DYNAMICS

The first obvious observation for the flight phase is the conservation of angular momentum. As the contact force f_c is a zero vector, there is no external force other than gravity acting on the system, i.e., $\dot{k}_G = \mathbf{0}_{3 \times 1}$. Hence, during the flight phase, the body orientation cannot directly be controlled by the actuators. It can only be controlled by the coupling effects of the actuated links and this information is embedded inside the A_j matrix. Define $k_{Gf} \in \mathbb{R}^3$ to be the centroidal angular momentum of the system during the flight phase such that:

$$k_{Gf} = [A_v \quad A_\omega \quad A_j] \begin{bmatrix} v_b \\ \omega_b \\ \nu_j \end{bmatrix}.$$

Due to the conservation of angular momentum, k_{Gf} is constant throughout the flight phase. The consequent body orientation dynamics yields:

$$\omega_b = A_\omega^{-1}(k_{Gf} - A_v v_b - A_j \nu_j),$$

where A_ω is always invertible by Property 2. Furthermore, due to Property 1, the body orientation dynamics results in

$$\omega_b = A_\omega^{-1}(k_{Gf} - A_j \nu_j). \quad (13)$$

The body orientation dynamics (13) implies a few important aspects. First and most apparent, it shows that the body

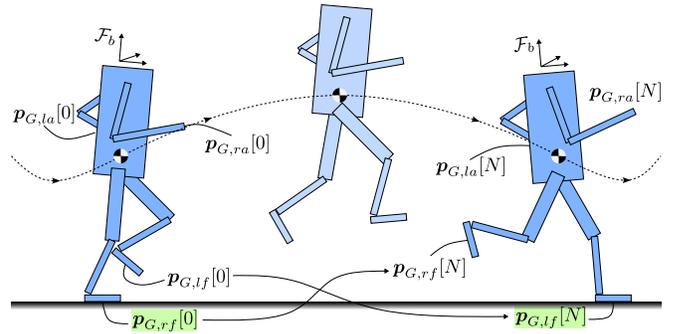


Fig. 1. Illustration of humanoid running from liftoff to the consecutive touchdown. The green variables (liftoff and touchdown point w.r.t. CoM) are given by any running or jumping algorithm and assumed to be known. Uncolored parameters are not captured by the reduced models and require manual selections. The objective of optimization is to eliminate these selections.

orientation can be implicitly controlled through the actuated link joints. Second, during the flight phase, the orientation dynamics is completely decoupled from the translational trajectory. This decoupling allows the body rotational velocity estimation to be independent of the robotic system's flight phase trajectory.

V. OPTIMIZATION PROBLEM FORMULATION

As discussed in the introduction, centroidal model-based running and jumping planners fall short of identifying how the body, arm, and swing leg trajectories should evolve. They usually capture only the essence of the locomotion, i.e., center of mass, stance foot, and contact force evolution of the system. However, as shown in Fig. 1, humanoid robots require much more to control. The stance foot location just before the liftoff phase is known. On the other hand, since the stance leg becomes the next step's swing leg, the user has to define an appropriate trajectory for its evolution. The same is valid for the swing leg. At the liftoff moment, the swing leg configuration is unknown and must be decided manually. However, at the next touchdown moment, the same leg becomes the stance leg and should be placed at a known location with respect to the center of mass. Nothing is given for arm evolution. They should be used to regulate the robot's posture. This section works on these unknown aspects and formulates an optimization problem to decide the limb evolution of the system.

Define $q^d : \mathbb{R} \rightarrow \mathbb{R}^n$ to be a vector of polynomials for the desired actuated joint evolution of the system for the flight phase. Hence, $q^d(0)$ and $q^d(t_f)$ represent the desired joint position at the beginning and end of the flight phase, respectively. Similarly, define θ_b to be the body orientation. Hence, for a given initial body orientation $\theta_b(0)$ the consequent final touchdown configuration integrated:

$$\theta_b(t_f) = \theta_b(0) + \int_0^{t_f} T(\theta_b(t), \omega_b(t)) dt, \quad (14)$$

where function T is a mapping from the angular velocities to the body configuration rate (for example Euler or Quaternion rate).

Algorithm 1 Integration of base orientation

```

1:  $\theta \leftarrow \theta(0)$ 
2:  $t \leftarrow 0$ 
3:  $\omega_b \leftarrow \omega_b(0)$ 
4:  $(\mathbf{A}_k, \mathbf{p}_G) = \text{computeCentroidalMap}(\theta_0, \mathbf{q}^d(0))$ 
5:  $\mathbf{k}_{Gf} = \mathbf{A}_k \nu$ 
6: while  $t < t_f$  do
7:    $\theta \leftarrow \theta + T(\omega_b(t)) * t_f/N$ 
8:    $t \leftarrow t + t_f/N$ 
9:    $(\mathbf{A}_k, \mathbf{p}_G) = \text{computeCentroidalMap}(\theta, \mathbf{q}^d(t))$ 
10:   $\omega_b = \mathbf{A}_\omega^{-1}(\mathbf{k}_{Gf} - \mathbf{A}_j \dot{\mathbf{q}}^d(t))$ 
11: end while
12:  $\omega_b(t_f) = \omega_b$ 

```

An algorithm that performs a discrete summation of the nonlinear integrator (14) through N number of sampling points is shown in Alg. 1. Starting from a given (desired) liftoff base orientation and rotational velocity, the algorithm integrates for the touchdown configuration. The “computeCentroidalMap” function in the algorithm calculates the forward kinematics, center of mass location, and centroidal momentum matrix together (see documentation of Pinochio [19] for “computeCentroidalMap”). The given algorithm will construct a nonlinear cost function for the optimization.

The computation cost of the optimization can be reduced via some reasonable assumptions. Typical leg joints of a humanoid robot are shown in Fig. 2. The ankle link is a relatively small component of the robot and constitutes an insignificant portion of the total inertia. Since its body is a small mass with a very short link length, rotation of ankle joints has a negligible effect on the overall inertia shape and angular momentum. Hence, during the optimization, these joints can be assumed not to move, i.e., remain at a constant angle. This assumption still accounts for ankle inertia and mass but neglects the inertia change with respect to the ankle joint angle. Similarly, the hip yaw joint is also another joint that can be simplified. Even though this joint drives the whole leg, as the mass is distributed around its rotation axis, the inertia that this joint drives is still comparably small. Another important aspect of this joint is that, as seen in Fig. 1, this joint is not very active during running and jumping but only during a change of direction. Hence, assuming this joint will stay at its default position during the flight phase is also a reasonable assumption. During the direction change step, where the support leg rotates to the new heading angle during the flight phase, its rotational effect can be accounted for in the optimization. Similar justifications can also be made for the arm joints, e.g., wrist and shoulder yaw joints. On the other hand, hip roll, hip pitch, shoulder pitch, and knee joints are highly inertial and have an important effect on the overall inertial shape. Additionally, these joints move in a wide range at high velocities and generate a significant portion of angular momentum. Hence, these joints are the main focus of the optimization

Let $\mathbf{\Gamma} \in \mathbb{R}^{n \times (m+1)}$ be the collection of degree m polyno-

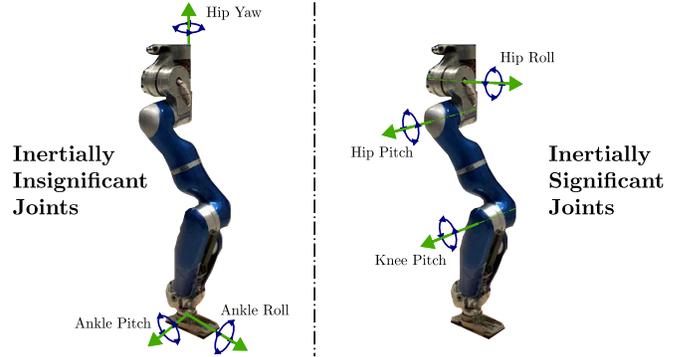


Fig. 2. Leg joints of a typical humanoid robot. The left figure shows the joints that do not have significant effects on the inertial shape or angular momentum of the robot. On the other hand, the right figure shows the joints that both translate and rotate significant portions of inertia. As an example, the image shows TORO’s right leg [20] and the joint locations are placed to illustrate the most common and intuitive configurations.

mial constants for all desired joint trajectories such that

$$\mathbf{q}^d = \underbrace{\begin{bmatrix} {}^1\gamma_m & {}^1\gamma_{m-1} & \cdots & {}^1\gamma_1 & {}^1\gamma_0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ {}^n\gamma_m & {}^n\gamma_{m-1} & \cdots & {}^n\gamma_1 & {}^n\gamma_0 \end{bmatrix}}_{\mathbf{\Gamma}} \underbrace{\begin{bmatrix} t^m \\ t^{m-1} \\ \vdots \\ t \\ 1 \end{bmatrix}}_t. \quad (15)$$

From this point, we label the upcoming touchdown leg (left foot in Fig. 1) and the upcoming swing leg (right foot in Fig. 1) as stance and swing foot, respectively.

A. Cost Function

The objective of the cost function is to minimize the body orientation at the next touchdown moment:

$$\min_{\mathbf{\Gamma}} f_{\text{cost}} = \|\theta_b(t_f)\|. \quad (16)$$

From a given liftoff body orientation, the optimizer modifies the joint trajectories through the polynomial gains such that there is minimal body orientation at the next touchdown. The base orientation throughout the flight phase in (16) can be calculated through Algorithm 1.

B. Constraints

- 1) Setting ankle, wrist, hip yaw, and arm yaw trajectories to zero polynomials:

$$\begin{bmatrix} \text{ankle } \mathbf{\Gamma} \\ \text{wrist } \mathbf{\Gamma} \\ \text{hipYaw } \mathbf{\Gamma} \\ \text{armYaw } \mathbf{\Gamma} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{4 \times 4} \\ \mathbf{0}_{4 \times 4} \\ \mathbf{0}_{2 \times 4} \\ \mathbf{0}_{2 \times 4} \end{bmatrix}. \quad (17)$$

In case of a direction change, during the transient step, the hip yaw joint trajectory of the related leg can be encoded here to take account of the inertial shape changes.

- 2) Enforcing the touchdown leg position for the upcoming stance phase through a forward kinematic constraint:

$$\mathbf{p}(\text{stance } \mathbf{\Gamma} t(t_f)) - \mathbf{p}_G(t_f) = \mathbf{p}_{G, \text{stance}}^d. \quad (18)$$

- 3) Enforcing the liftoff swing leg position through a forward kinematic constraint:

$$\mathbf{p}^{(\text{swing} \Gamma \mathbf{t}(0))} - \mathbf{p}_G(t_f) = \mathbf{p}_{G, \text{swing}}^d. \quad (19)$$

- 4) Enforcing the touchdown stance leg relative velocity to zero through a forward kinematic constraint:

$$\dot{\mathbf{p}}^{(\text{stance} \Gamma \mathbf{t}(t_f), \text{stance} \Gamma \dot{\mathbf{t}}(t_f))} - \dot{\mathbf{p}}_G(t_f) = \mathbf{0}. \quad (20)$$

- 5) Enforcing the liftoff swing leg velocity to zero:

$$\dot{\mathbf{p}}^{(\text{swing} \Gamma \mathbf{t}(0), \text{swing} \Gamma \dot{\mathbf{t}}(0))} = \mathbf{0}. \quad (21)$$

- 6) Minimum or desired clearance between the next stance leg and the ground at the beginning of the flight phase:

$$\mathbf{p}^{(\text{stance} \Gamma \mathbf{t}(0))} = h_{\text{stance}}. \quad (22)$$

- 7) Minimum or desired clearance between the next swing leg and the ground at the end of the flight phase:

$$\mathbf{p}^{(\text{swing} \Gamma \mathbf{t}(t_f))} = h_{\text{swing}}. \quad (23)$$

VI. SIMULATION RESULTS

The simulation section includes verification of the proposed method through two different humanoid robots. We present the optimization results and then show the optimized trajectories' behavior through a running algorithm. As this study composes a subpart of humanoid running or jumping (see Fig. 3), we inherit a SLIP-based trajectory generation method from [9]. The running algorithm is combined with the flight phase limb swing trajectories generated by the proposed optimization problem. The overview of the nonlinear optimization solver algorithm that we implemented in C++ can be found in Appendix.

A. Results on Kangaroo

We first present the optimization result on the Kangaroo bipedal robot (see Fig. 4). Swing leg trajectories of the robot are particularly important as it cannot take advantage of arm swinging for regulation purposes. A running trajectory with $1m/s$ forward velocity, $0.24s$ stance time, and 0.31 seconds flight time ($\approx 12cm$ of vertical jumping) is obtained from [9] and the optimization problem is configured with the flight time, CoM liftoff velocity, and the relative stance foot locations. It is worth noticing the length of the flight phase of the generated trajectory as it makes postural control harder and causes longer error accumulation in case

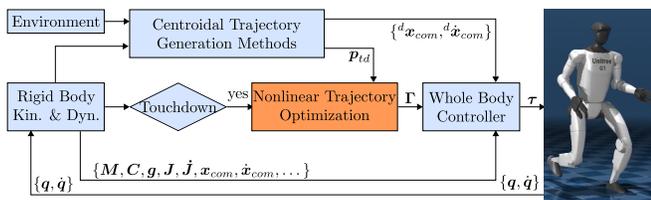


Fig. 3. A generic control system diagram of humanoid running or jumping through centroidal trajectory generation methods. This study fits into the orange block which is triggered at the touchdown moment and determines the limb swing trajectories for the flight phase.

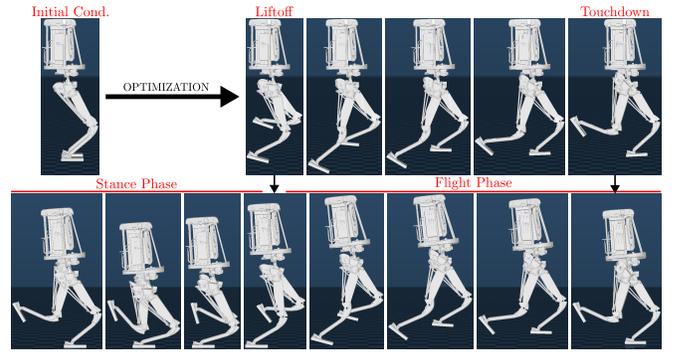


Fig. 4. Top row: optimization playback of the flight phase. Bottom row: simulation verification of the optimized trajectories on a running algorithm.

of imprecision. A 3rd-degree polynomial is employed for each inertially significant joint trajectory. The optimization problem includes 24 optimization parameters along with the 14 nonlinear constraints and samples of 11 points in the dynamics. The optimization problem takes $1.54ms$ to solve on a daily use desktop computer with AMD Ryzen 7 5800X CPU.

The snapshots of the optimized limb trajectories are shown in the top row of Fig. 4. The optimization result suggests that if the robotic system liftoffs with the optimized configuration and velocities and follows the given flight phase swinging behavior, the next touchdown will happen with minimal torso orientation. The verification of the optimized trajectories on the running simulation is shown in the bottom row of Fig. 4. The optimization problem is triggered at the moment of touchdown. During the stance phase, the whole-body controller is commanded to liftoff with the optimized configuration and velocities. During the flight phase, the optimized trajectories are followed. The only difference between the optimization playback and the verification simulation is the ankle configuration. Due to its negligible inertial effects, the optimization problem assumes the ankle is always at zero configuration. On the other hand, they are adequately actuated in the simulation. The simulation shows almost identical results with the optimization playback. The optimization playback and running simulation animations are presented in the supplemental video.

The first thing to notice on the optimized trajectory is the swing-back behavior of the liftoff leg. This behavior is performed to balance the centroidal angular momentum in the sagittal plane so that the torso movement remains minimal. In order to reason the optimization results, we plot the centroidal angular momentum portions of each limb in the sagittal plane in Fig. 5. The flight phase angular momentum is also an implicit product of the optimization problem and is constant due to the conservation. As the left leg moves forward, it generates a negative momentum on the CoM frame. The figure shows that the swing-back behavior of the right leg generates a positive momentum to cancel the effect of the other leg's swing. As the left leg reaches its desired position, its velocity and momentum contribution fade away. As the optimization objective is to

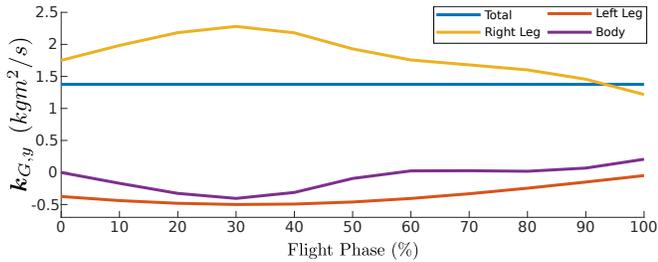


Fig. 5. The total centroidal angular momentum in the sagittal plane along with the contribution of each limb. The figure shows that the limb swing trajectories keep the angular momentum portion of the body around zero.

minimize the body orientation for the next touchdown, the right leg covers most of the angular momentum and keeps the remaining momentum for the torso around zero. In the case of a tilted liftoff condition, the limbs do not cancel the whole momentum and let the body rotate back to the minimal body orientation. Such behavior can also be used for disturbance rejection purposes.

B. Results on Unitree G1

In order to show the generalizability and inclusiveness of the proposed method, we also optimize for and simulate Unitree’s G1 humanoid robot. A running trajectory with 1m/s forward velocity, 0.21s stance time, and 0.26 seconds flight time is generated, and the optimization problem is configured with the flight time, CoM liftoff velocity, and the relative stance foot locations. A 3rd-degree polynomial is employed for each inertially significant leg joint and shoulder pitch joint. The optimization problem includes 32 (24+8) optimization parameters along with the 14 nonlinear constraints and samples of 11 points in the dynamics and takes 1.92ms to solve. The optimization playback and running simulation animations are presented in the supplemental video.

A similar swing-back behavior on the right leg can be observed in Fig. 6. Differently, the additional arm swing is apparent in the movement. As the right leg covers most of the sagittal centroidal angular momentum, the arm movements balance the body rotation in the vertical axis (transverse plane). Even though the optimization is initiated with the zero polynomial configuration, it still manages to converge.

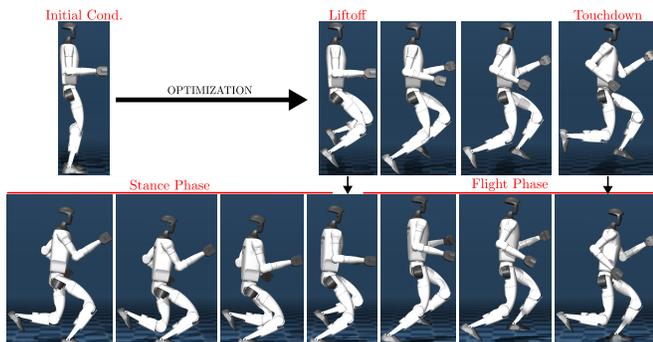


Fig. 6. Top row: optimization playback of the flight phase. Bottom row: simulation verification of the optimized trajectories on a running algorithm.

VII. CONCLUSION

This study addresses the problem of determining limb swing trajectories for the flight phases of humanoid running. We propose a nonlinear optimization problem to find a set of proper limb swing trajectories that place the stance leg at the desired location and keep the body upright for the next stance phase. We first explore some properties of the angular momentum portion of the centroidal momentum matrix. Then, taking advantage of these properties, we construct a cost function with some constraints for foot placement points. We show that the size of the optimization problem can be significantly reduced through some mild simplifications. We implement the optimized trajectories on a running algorithm from the literature for verification and perform simulations on two different robots. The simulation results verify that once the robotic system lifts off with the optimized configuration and velocities and tracks the optimized flight phase limb trajectories, the robotic system lands with a proper and minimal body orientation. We also share the computational performance results and verify that the proposed optimization problem can be solved in real-time. Lastly, we provide an overview algorithm for the solver that we implement to solve the optimization problem.

VIII. APPENDIX: SOLVER ALGORITHM

We implement our solver, for the given nonlinear optimization problem. For the completeness of the paper, an overview of the solver with the equality constraints is provided in Algorithm 2. The detailed algorithms can be found in [21]. The tool is available at

<https://github.com/ssovukluk/ENFORCcpp> (★)

Algorithm 2 Overview of the nonlinear optimizer with equality constraints.

- 1) Define n_c to be the number of constraints, $f_{c,i}$ to be the i^{th} constraint function, and \mathbf{J} to be an empty matrix to collect constraint function gradients one by one.
 - 2) If $n_c \neq 0$, set $i = 1$. Otherwise, proceed to step 8.
 - 3) Calculate numerical gradient of $f_{c,i}$ w.r.t. the optimization variables.
 - 4) Store the gradient vector: $\mathbf{J}.\text{col}(i) = \nabla f_{c,i}$.
 - 5) If $i > 1$, project the estimated gradient into the Nullspace of the vector space \mathbf{J} such that iterating with this gradient does not disturb the previous constraints.
 - 6) Implement a line search algorithm to find the zero crossing point of the equality constraint via iterating through the (projected) gradient.
 - 7) If $i < n_c$, set $i = i + 1$ and go to step 3.
 - * At this point, all the equality constraints are solved, and the optimizer is ready to proceed to minimize the cost function.
 - 8) Calculate the numerical gradient of the cost function f_{cost} .
 - 9) Project the gradient ∇f_{cost} into the Nullspace of the vector space \mathbf{J} such that iterating with this gradient does not disturb the equality constraints.
 - 10) Implement a line search algorithm to find the local minimum.
 - 11) Check all the termination conditions and go to step 2 if none of them is triggered.
-

REFERENCES

- [1] D. E. Orin, A. Goswami, and S.-H. Lee, "Centroidal dynamics of a humanoid robot," *Autonomous robots*, vol. 35, pp. 161–176, 2013.
- [2] P.-B. Wieber, "Holonomy and nonholonomy in the dynamics of articulated motion," in *Fast Motions in Biomechanics and Robotics: Optimization and Feedback Control*. Springer, 2006, pp. 411–425.
- [3] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: A step toward humanoid push recovery," in *2006 6th IEEE-RAS International Conference on Humanoid Robots*, 2006, pp. 200–207.
- [4] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," in *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, vol. 2, 2003, pp. 1620–1626 vol.2.
- [5] J. Engelsberger, C. Ott, and A. Albu-Schäffer, "Three-dimensional bipedal walking control based on divergent component of motion," *IEEE Transactions on Robotics*, vol. 31, no. 2, pp. 355–368, 2015.
- [6] H. Geyer, A. Seyfarth, and R. Blickhan, "Compliant leg behaviour explains basic dynamics of walking and running," *Proceedings of the Royal Society B: Biological Sciences*, vol. 273, no. 1603, pp. 2861–2867, 2006.
- [7] C. Hubicki, A. Abate, P. Clary, S. Rezazadeh, M. Jones, A. Peekema, J. Van Why, R. Domres, A. Wu, W. Martin, H. Geyer, and J. Hurst, "Walking and running with passive compliance: Lessons from engineering: A live demonstration of the atrias biped," *IEEE Robotics & Automation Magazine*, vol. 25, no. 3, pp. 23–39, 2018.
- [8] P. M. Wensing and D. E. Orin, "High-speed humanoid running through control with a 3d-slip model," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 5134–5140.
- [9] S. Sovukluk, J. Engelsberger, and C. Ott, "Highly maneuverable humanoid running via 3d slip+foot dynamics," *IEEE Robotics and Automation Letters*, vol. 9, no. 2, pp. 1131–1138, 2024, doi:10.1109/LRA.2023.3342668.
- [10] J. Engelsberger, P. Kozłowski, C. Ott, and A. Albu-Schäffer, "Biologically inspired deadbeat control for running: From human analysis to humanoid control and back," *IEEE Transactions on Robotics*, vol. 32, no. 4, pp. 854–867, 2016.
- [11] S. Sovukluk, J. Engelsberger, and C. Ott, "Whole body control formulation for humanoid robots with closed/parallel kinematic chains: Kangaroo case study," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 10390–10396, doi:10.1109/IROS55552.2023.10341391.
- [12] X. Xiong and A. D. Ames, "Sequential motion planning for bipedal somersault via flywheel slip and momentum transmission with task space control," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 3510–3517.
- [13] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, "Dynamic locomotion in the mit cheetah 3 through convex model-predictive control," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 1–9.
- [14] P. M. Wensing, M. Posa, Y. Hu, A. Escande, N. Mansard, and A. D. Prete, "Optimization-based control for dynamic legged robots," *IEEE Transactions on Robotics*, vol. 40, pp. 43–63, 2024.
- [15] M. M. Sotaro Katayama and Y. Tazaki, "Model predictive control of legged and humanoid robots: models and algorithms," *Advanced Robotics*, vol. 37, no. 5, pp. 298–315, 2023.
- [16] R. Schuller, G. Mesesan, J. Engelsberger, J. Lee, and C. Ott, "Online centroidal angular momentum reference generation and motion optimization for humanoid push recovery," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5689–5696, 2021.
- [17] P. M. Wensing and D. E. Orin, "Improved computation of the humanoid centroidal dynamics and application for whole-body control," *International Journal of Humanoid Robotics*, vol. 13, no. 01, p. 1550039, 2016.
- [18] R. Featherstone and D. E. Orin, *Dynamics*. Cham: Springer International Publishing, 2016, pp. 37–66.
- [19] J. Carpentier, F. Valenza, N. Mansard *et al.*, "Pinocchio: fast forward and inverse dynamics for poly-articulated systems," <https://stack-of-tasks.github.io/pinocchio>, 2015–2021.
- [20] J. Engelsberger, A. Werner, C. Ott, B. Henze, M. A. Roa, G. Garofalo, R. Burger, A. Beyer, O. Eiberger, K. Schmid, and A. Albu-Schäffer, "Overview of the torque-controlled humanoid robot toro," in *2014 IEEE-RAS International Conference on Humanoid Robots*, 2014, pp. 916–923.
- [21] S. Sovukluk and C. Ott, "An efficient numerical function optimization framework for constrained nonlinear robotic problems," *arXiv preprint arXiv:2501.17349*, 2025, doi:10.48550/arXiv.2501.17349.