



Article

Single Sequential Trajectory Optimization with Centroidal Dynamics and Whole-Body Kinematics for Vertical Jump of Humanoid Robot

Yaliang Liu ¹, Xuechao Chen ^{1,2,*} , Zhangguo Yu ^{1,2} , Haoxiang Qi ¹ and Chuanku Yi ¹

¹ School of Mechatronical Engineering, Beijing Institute of Technology, Beijing 100081, China; liuyaliang@bit.edu.cn (Y.L.); yuzg@bit.edu.cn (Z.Y.); 3120215098@bit.edu.cn (H.Q.); yichuanku@bit.edu.cn (C.Y.)

² Key Laboratory of Biomimetic Robots and Systems, Ministry of Education, Beijing 100081, China

* Correspondence: chenxuechao@bit.edu.cn

Abstract: High vertical jumping motion, which enables a humanoid robot to leap over obstacles, is a direct reflection of its extreme motion capabilities. This article proposes a single sequential kinodynamic trajectory optimization method to solve the whole-body motion trajectory for high vertical jumping motion. The trajectory optimization process is decomposed into two sequential optimization parts: optimization computation of centroidal dynamics and coherent whole-body kinematics. Both optimization problems converge on the common variables (the center of mass, momentum, and foot position) using cost functions while allowing for some tolerance in the consistency of the foot position. Additionally, complementarity conditions and a pre-defined contact sequence are implemented to constrain the contact force and foot position during the launching and flight phases. The whole-body trajectory, including the launching and flight phases, can be efficiently solved by a single sequential optimization, which is an efficient solution for the vertical jumping motion. Finally, the whole-body trajectory generated by the proposed optimized method is demonstrated on a real humanoid robot platform, and a vertical jumping motion of 0.5 m in height (foot lifting distance) is achieved.

Keywords: humanoid robot; vertical jump; trajectory optimization



Citation: Liu, Y.; Chen, X.; Yu, Z.; Qi, H.; Yi, C. Single Sequential Trajectory Optimization with Centroidal Dynamics and Whole-Body Kinematics for Vertical Jump of Humanoid Robot. *Biomimetics* **2024**, *9*, 274. <https://doi.org/10.3390/biomimetics9050274>

Academic Editor: Jinyou Shao

Received: 8 April 2024

Revised: 25 April 2024

Accepted: 28 April 2024

Published: 2 May 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Humanoid robotics is a burgeoning research field that has great potential for technological advancement and human prosperity. Considerable research has been conducted on the walking and manipulation of humanoid robots, which enables a robot to move through complex terrains and perform dangerous and repetitive tasks [1–4]. To further explore the full potential of humanoid robots and equip them with the ability to execute highly dynamic tasks, researchers are delving deeper into the study of highly dynamic motions, such as vertical jumping.

A high vertical jump is a challenging task, and many researchers have been working on jumping motions using simple mechanisms or simulations [5–12]. In the last few years, few have realized vertical jumping of a real full-scale humanoid robot platform [13–18]. Apart from excellent actuation hardware optimization [14,15,19], dynamic motion control is the key to the success of vertical jumping. Among the current jumping motion control frameworks [13,15–18], a whole-body trajectory can generate an effective jumping motion during the launching phase [6,7,10,12,16,17], and online control algorithms are designed to compensate as much as possible for some motion errors caused by an inaccurate trajectory. Thus, a whole-body trajectory that satisfies whole-body dynamics constraints can reduce the design complexity of the control algorithms for vertical jump. Because of the complexity of the motions and the high degrees of freedom of a humanoid robot, trajectory optimization has emerged as the most widely used method for generating specified motion trajectories.

Trajectory optimization methods are not limited to the field of humanoid robot jumping and can be divided into three categories. (1) Direct use of the complete whole-body dynamics model for planning: This enables the robot to produce satisfactory trajectories for more complex motion tasks [20]. However, this method sometimes becomes intractable, especially when dealing with high-dimensional, complex robot models. (2) Reduced-order models, such as the spring-loaded inverted pendulum (SLIP), flywheel SLIP, and single rigid body model (SRBM), are used to plan jumping motions. Despite the success of SLIP, its point mass model completely ignores angular momentum [16,21]. Jump motions often involve body rotation, for which angular momentum is indispensable. To alleviate this issue, flywheel SLIP is a potential solution [12]. Furthermore, the assumption of massless legs enables the SRBM to successfully render gait and motion generation problems computationally tractable for quadruped robots [22–24]. Unfortunately, humanoid robots have more actuated joints in each leg than a quadruped robot, and hence, the assumption of massless legs is easily violated. (3) Another simplified model, known as kino-dynamic [25], uses centroidal dynamics to reduce the complexity of the whole-body dynamics while guaranteeing feasible kinematic motions. Centroidal dynamics efficiently introduce the angular momentum, which is beneficial for the generation of arbitrary jumping motions. Recently, an alternating kino-dynamic optimization planner has been proposed that iteratively solves two optimization subproblems (the centroidal dynamics and kinematics optimization) until they reach consensus, allowing it to effectively solve whole-body motions [10,26,27].

Specifically, alternating kino-dynamic optimization makes it easier to find a locally optimal solution. During one iteration process, one subproblem needs to perform tracking tasks for common variables (the center of mass (CoM); momentum, including both the angular and linear momentum; and foot position) that were optimized by the other subproblem. Each tracking task is treated equally and objectively. Therefore, it has to iterate many times to ensure the consistency of common variables, which is obviously time-consuming. For example, Budhiraja et al. [27] obtained the whole-body trajectory with high consistency using centroidal dynamics and kinematics after 10 iterations. Based on their work, we propose a single sequential kino-dynamic trajectory optimization method that is designed on the basis of an alternating kino-dynamic planner. This method adds cost functions to ensure that both subproblems come to an agreement on the common variables (CoM and momentum), but consistency in the foot position is not required. In this case, the whole-body trajectory is effectively generated by a single sequential optimization, which first solves the centroidal dynamics optimization and then solve the whole-body kinematics optimization, and it is undoubtedly faster than the alternating kino-dynamic optimization planner. Obviously, the optimization solution time will be further reduced so that the whole-body trajectory of jumping motions can be optimized online, and the jumping flexibility of the robot can be further improved.

In addition, the above studies [16,28] focused only on the trajectory optimization of the launching or landing phases, paying little attention to the flight phase. However, to achieve a high jump, the active retraction motion of the leg is essential in the flight phase, which is a problem that remains to be solved. More importantly, when searching for the optimal motions for the launching phase and flight phase, it is necessary to know whether the foot is touching the environment and whether any forces are present. In other words, deciding which of the feet should be in contact at a given time is a matter of finding the contact schedule. A traditional approach is to use a soft-contact model, such as a spring–damper contact model [29], to approximate inherently hard contact surfaces. These virtual spring–damper models need to be exceptionally stiff to emulate a real surface. However, the abrupt contact force changes resulting from a foot hitting a rigid surface can impede the optimizer’s performance. Mixed-integer programming has proved to be an efficient tool to efficiently solve contact schedules, gait transitions, and motion [30,31]. It involves assigning contact to surfaces using mixed integers and can be combined with convex or mixed-integer convex dynamic models, and hence, the computation is large and complicated. In this article, the contact sequence can be defined in advance because a

jumping motion with contact is relatively simple. Then, the contact sequence is combined with the linear complementarity condition [24,25,32], which can enforce that either the foot must be zero distance from the contact surface or the force must be zero.

In order to solve the above problems and obtain the whole-body trajectory of a vertical jumping motion, including the launching and flight phases, a single sequential kino-dynamic trajectory optimization method is proposed in this work. The brief structure of the optimization method is illustrated in Figure 1. As shown, the task description in the optimization framework includes the initial robot pose (CoM, joint position, and foot position) and final robot pose estimation, motion duration, knot point and re-specified contact sequence, which are important initialization variables for optimization problems. After inputting the initial robot pose, knot point, and other variables into the centroidal dynamics optimization, a set of CoM and momentum values are obtained that satisfy the centroidal dynamics and other feasible sets of dynamic constraints while minimizing the cost function $\Phi^{\text{cen}}(\cdot)$. The optimized CoM, momentum, and foot position are used as reference inputs for the whole-body kinematics optimization, and the initialization variables such as initial joint angle, the knot points, and other variables in the task description are also required. Then, the whole-body kinematics optimization is solved under the above inputs and kinematic constraints. More importantly, the cost function $\Phi^{\text{kin}}(\cdot)$ is deliberately added to ensure that both optimization problems come to an agreement on the common variables (CoM and momentum) without requiring consistency in the foot position. Moreover, cost functions $\phi^{\text{cen}}(\cdot)$ and $\phi^{\text{kin}}(\cdot)$ are used to penalize the state variables to discourage violent motions. Finally, a whole-body motion trajectory is obtained that satisfies both the centroidal dynamic and whole-body kinematic constraints, enabling the humanoid robot to achieve a high vertical jump. As for the landing phase, the landing controller designed for our team’s previous research [17] is adopted to ensure landing stability. The CoM trajectory that is solved by centroidal dynamics optimization is used as the reference input for the landing controller.

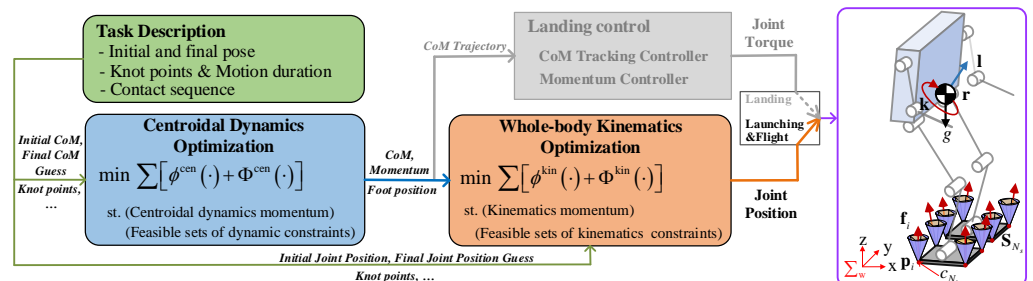


Figure 1. Single sequential kino-dynamic trajectory optimization framework: $\phi^{\text{cen}}(\cdot)$, $\Phi^{\text{cen}}(\cdot)$, $\phi^{\text{kin}}(\cdot)$ and $\Phi^{\text{kin}}(\cdot)$ represent different cost functions. The contact surface of the foot is S_{N_s} , c_{N_i} denotes the vertex of the contact surface S_{N_s} , $p_i \in \mathbb{R}^3$ denotes the position of contact point c_{N_i} , and $f_i \in \mathbb{R}^3$ is a contact force acting on contact point c_{N_i} . Here, f_i is restricted to the inside of the friction cone (shown in purple) or a more restrictive friction pyramid (shown in yellow), r represents the CoM position, the centroidal angular momentum of the robot is k , the centroidal linear momentum of the robot is l , $g \in \mathbb{R}^3$ is the gravity vector, and \sum_w denotes the world coordinate.

The contribution of this work is summarized as follows:

- (1) A single sequential kino-dynamic trajectory optimization framework is proposed to solve the optimal jumping motion problem. The whole-body trajectory is effectively generated by a single sequential optimization, which is an efficient solution.
- (2) This optimization framework can generate vertical jumping motions with launching and flight phases, which are essential for highly dynamic motions.

This paper is organized as follows. The single sequential kino-dynamic trajectory optimization method is introduced in Section 2. Section 3 discusses the simulation and experimental results. Finally, the conclusion is presented in Section 4.

2. Single Sequential Kino-Dynamic Trajectory Optimization

In this section, the single sequential kino-dynamic trajectory optimization method is introduced. The centroidal dynamics optimization and whole-body kinematics optimization are described separately.

2.1. Centroidal Dynamics Optimization

Suppose the contact surface between a humanoid robot (i.e., the robot model shown in Figure 1) and its environment is described by S_{N_s} , which is a convex polygon with vertices $(c_{N_1}, \dots, c_{N_i}, \dots, c_{N_c})$ [33,34]. Moreover, the robot's state can be represented by centroidal linear and angular momentums, and the rates of the linear and angular momentums are determined by the total external force and gravity. Therefore, the equations of motion of the whole robot are given by:

$$\dot{\mathbf{h}} = \begin{bmatrix} \dot{\mathbf{l}} \\ \dot{\mathbf{k}} \end{bmatrix} = \begin{bmatrix} m\mathbf{g} \\ \mathbf{0} \end{bmatrix} + \sum_{i=1}^{N_c} \begin{bmatrix} \mathbf{f}_i \\ (\mathbf{p}_i - \mathbf{r}) \times \mathbf{f}_i \end{bmatrix} \quad (1)$$

where m represents the mass of the robot.

Generally, researchers use the centroidal dynamics model (Equation (1)) to optimize the CoM and momentum and then combine it with online control algorithms to realize the desired motion [28,35]. To satisfy the real-time demands of an online algorithm, the above optimization approach prioritizes efficiency, often at the cost of trajectory accuracy. In our proposed framework, the centroidal trajectory is used as a reference input for the whole-body kinematic optimization. However, any error in the centroidal trajectory tends to become amplified when mapping the whole-body kinematics. This undoubtedly has a negative impact on highly dynamic jumping motions. Throughout this article, the higher-order dynamic expression of Equation (1) is referred to as the centroidal dynamics model to enhance trajectory accuracy.

The optimization variables for centroidal dynamics optimization include the position \mathbf{r} , velocity $\dot{\mathbf{r}}$, and acceleration $\ddot{\mathbf{r}}$ of the robot's CoM; the angular momentum \mathbf{k} and its time derivative $\dot{\mathbf{k}}$; the linear momentum \mathbf{l} and its time derivative $\dot{\mathbf{l}}$; and the contact force \mathbf{f}_i and its time derivative $\dot{\mathbf{f}}_i$. Here, we sample all these time-varying quantities at the knot point N , with time duration T between successive knot points. Additionally, the contact sequence \mathbf{c}_i is pre-specified, which dictates the sequence in which contact points make and break contact with the ground (touch \rightarrow off).

The centroidal dynamics optimization can be posed in the following manner:

$$\min_{\mathbf{h}, \mathbf{p}_i} \sum_{k=1}^N [\phi_k^{\text{cen}}(\cdot) + \Phi_k^{\text{cen}}(\cdot)] \quad (2a)$$

$$\begin{bmatrix} \ddot{\mathbf{r}}[k] \\ \dot{\mathbf{l}}[k] \\ \dot{\mathbf{k}}[k] \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{l}}[k]/m \\ \sum_{i=1}^{N_c} \mathbf{f}_i[k] \\ \sum_{i=1}^{N_c} \begin{bmatrix} (\dot{\mathbf{p}}_i[k] - \dot{\mathbf{r}}[k]) \times \mathbf{f}_i[k] + \\ (\mathbf{p}_i[k] - \mathbf{r}[k]) \times \dot{\mathbf{f}}_i[k] \end{bmatrix} \end{bmatrix} \quad (2b)$$

$$\dot{\mathbf{r}}[k+1] - \dot{\mathbf{r}}[k] = \ddot{\mathbf{r}}[k]\Delta t, \mathbf{r}[k+1] - \mathbf{r}[k] = \dot{\mathbf{r}}[k]\Delta t \quad (2c)$$

$$\dot{\mathbf{l}}[k+1] - \dot{\mathbf{l}}[k] = \dot{\mathbf{l}}[k]\Delta t, \mathbf{l}[k+1] - \mathbf{l}[k] = \mathbf{l}[k]\Delta t \quad (2d)$$

$$\dot{\mathbf{k}}[k+1] - \dot{\mathbf{k}}[k] = \dot{\mathbf{k}}[k]\Delta t, \mathbf{k}[k+1] - \mathbf{k}[k] = \mathbf{k}[k]\Delta t \quad (2e)$$

$$\dot{\mathbf{f}}_i[k+1] - \dot{\mathbf{f}}_i[k] = \dot{\mathbf{f}}_i[k]\Delta t, \mathbf{f}_i[k+1] - \mathbf{f}_i[k] = \mathbf{f}_i[k]\Delta t \quad (2f)$$

$$\sqrt{(f_i^x[k])^2 + (f_i^y[k])^2} \leq \mu(f_i^z[k]), f_i^z[k] \geq 0 \quad (2g)$$

$$f_i^z \leq f_{\max} \mathbf{c}_i[k] \quad (2h)$$

$$\left(\mathbf{p}_i^{xy}[k+1] - \mathbf{p}_i^{xy}[k]\right) \mathbf{c}_i[k] = 0 \tag{2i}$$

$$\mathbf{p}_i^z[k] - h_{terrain} \geq 0 \tag{2j}$$

$$\ell^{\min} \leq \|\mathbf{p}[k] - \mathbf{r}[k]\|^2 \leq \ell^{\max} \tag{2k}$$

$$\mathcal{F}(\mathbf{p}_i[k]) \in \mathbf{S}_{N_s} \tag{2l}$$

The quadratic costs (see Equation (2a)) include a running cost $\phi_k^{\text{cen}}(\cdot)$ and consensus cost $\Phi_k^{\text{cen}}(\cdot)$. Running cost $\phi_k^{\text{cen}}(\cdot)$ is composed of user-defined tack costs, which penalize control variable $\ddot{\mathbf{f}}_i$ to discourage violent motions. It can be defined as follows:

$$\phi_k^{\text{cen}}(\cdot) = \sum_{i=1}^{N_c} \|\ddot{\mathbf{f}}_i[k]\|_{\mathbf{Q}_{cf}}^2 \tag{3}$$

where $\|\mathbf{x}\|_{\mathbf{Q}}^2$ is an abbreviation of quadratic cost $\mathbf{x}^T \mathbf{Q} \mathbf{x}$, $\mathbf{Q} \geq 0$, and $\mathbf{Q}_{cf} \in \mathbb{R}^{6 \times 6}$ is a force weight matrix. Additionally, square bracket $[k]$ indicates the sampled value at knot point k .

Additionally, consensus cost $\Phi_k^{\text{kin}}(\cdot)$ can penalize deviations in the CoM and momentum from the given references. It is expressed as follows:

$$\Phi_k^{\text{cen}}(\cdot) = \|\mathbf{r}[k] - \text{ref} \mathbf{r}[k]\|_{\mathbf{Q}_{cr}}^2 + \|\mathbf{h}[k] - \text{ref} \mathbf{h}[k]\|_{\mathbf{Q}_{ch}}^2 + \sum_{i=1}^{N_c} \left(\|\mathbf{r}[k] - \mathbf{p}_i[k]\|_{\mathbf{Q}_{ce}}^2 \right) \tag{4}$$

where $\text{ref} \mathbf{r}$ and $\text{ref} \mathbf{h}$ are the references obtained by linear interpolation from the initial value and final value estimate, respectively. $\mathbf{Q}_{cr} \in \mathbb{R}^{3 \times 3}$, $\mathbf{Q}_{ch} \in \mathbb{R}^{3 \times 3}$, and $\mathbf{Q}_{ce} \in \mathbb{R}^{3 \times 3}$ are weight matrices.

Centroidal dynamics optimization includes the following constraints:

Integration constraint : The constraints of the centroidal dynamics include dynamic (Equation (2b)) and integration constraints (Equation (2c,f)). Note that the forward-Euler integration [36] is adopted to approximate the time derivatives for the variables \mathbf{r} , $\dot{\mathbf{r}}$, $\dot{\mathbf{l}}$ and \mathbf{k} , $\dot{\mathbf{k}}$, \mathbf{f}_i , $\dot{\mathbf{f}}_i$, and Δt indicates time duration.

Frictional constraint: The contact force is limited to the inside of the friction cone (friction coefficient μ) to prevent the contact point from sliding, as expressed in Equation (2g). Additionally, the normal forces of the contact force have maximum limits during the launching phase and are always equal to zero during the flight phase. After combining the contact sequence, the constraint on the normal force is expressed by Equation (2h).

Terrain constraint: Referring to the complementarity constraint [24,25], we also impose a constraint that specifies that the foot in contact with the ground does not slide during the launching phase. The tangential displacement between two knot points must be zero when the foot is in contact. This constraint disappears during the flight phase, as shown in Equation (2i,j); $h_{terrain}$ represents the height of the ground.

Distance constraint: The distance constraint (see Equation (2k)) between the CoM and foot position should be within the predefined range ℓ^{\min} and ℓ^{\max} . The maximum value ℓ^{\max} is used to limit the overall length of the robot when stretched, which prevents some joints from approaching singularity. Similarly, the minimum value is used to prevent the robot from contracting too much during the flight phase.

Vertices constraint: The relative positions between the vertices (c_1, \dots, c_{N_c}) of the foot contact surface \mathbf{S}_{N_s} are fixed to form a virtual, fixed-shaped sole (Equation (2l)).

This optimization solves the momentum and contact forces in the launching and flight phases. However, it completely ignores the whole-body kinematics of the robot. How to map these to the whole-body trajectory after accurate centroidal trajectories have been obtained is worth researching.

2.2. Whole-Body Kinematics Optimization

In this section, the whole-body kinematics optimization is considered to be aligned with the centroidal dynamics optimization results, and the whole-body trajectory for the jumping motion is optimized.

Because both optimization problems want to be consistent in terms of common variables (CoM, momentum, and foot position), it is necessary to establish a kinematic model. This article mainly describes the calculation process of momentum, but the expressions of CoM and foot position are not described in detail. A humanoid robot can be modeled as a system consisting of $n + 1$ interconnected links and n joints, as shown in Figure 2c. The symbol Σ_w represents the world coordinate frame, and Σ_1 represents the float base coordinate frame. The left joints of the robot are represented as $\Sigma_{2\sim7}^L$ and Σ_{14}^L , and the right joints are represented as $\Sigma_{8\sim13}^R$ and Σ_{15}^R . Through the recursive relationship between link j and its predecessor link $p(j)$, the spatial speed of link j is expressed as follows:

$$\mathbf{v}_j = \begin{bmatrix} \omega_j \\ v_j \end{bmatrix} = {}^j\mathbf{X}_{p(j)} \mathbf{v}_{p(j)} + \Phi_j \dot{\mathbf{q}}_j \quad (5)$$

where ω_j and v_j represent the angular and linear velocities of link j , respectively, in the link coordinate frame; ${}^j\mathbf{X}_{p(j)}$ is a 6×6 spatial transform that converts spatial motion vectors from the predecessor $p(j)$ to j coordinates. The matrix Φ_j depends on the type of joint and has full column rank, $\dot{\mathbf{q}}_j$ signifies the velocity vector of link j relative to the velocity of its predecessor $p(j)$.

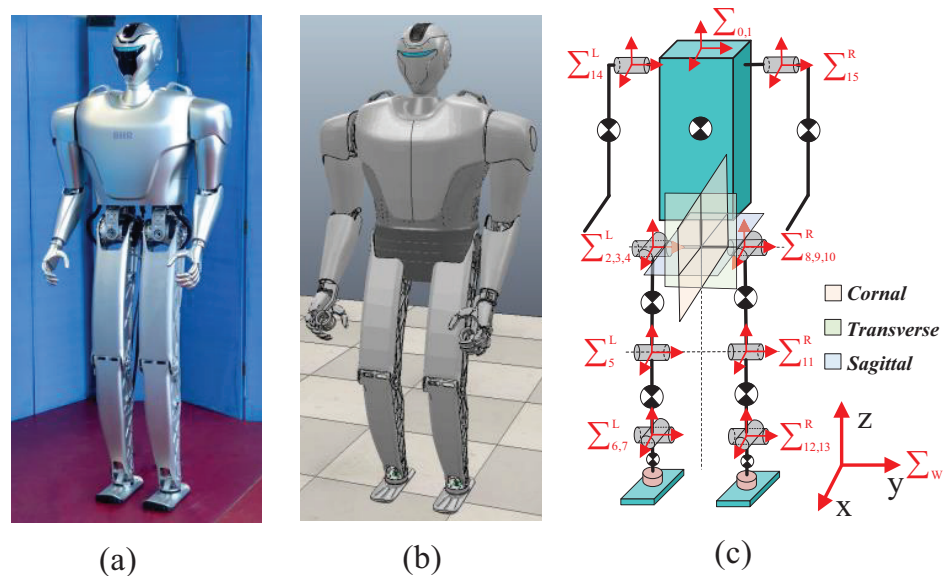


Figure 2. Robot platform used in this article. (a) Actual platform. (b) CoppeliaSim model. (c) Simplified link model.

The spatial transform ${}^j\mathbf{X}_{p(j)}$ can be composed from the position vector ${}^{p(j)}\mathbf{p}_j$, which originates from coordinate frame $p(j)$ to the j coordinate, and the 3×3 rotation matrix ${}^j\mathbf{R}_{p(j)}$ from coordinate frame $p(j)$ to j :

$${}^j\mathbf{X}_{p(j)} = \begin{bmatrix} {}^j\mathbf{R}_{p(j)} & \mathbf{0} \\ {}^j\mathbf{R}_{p(j)} \mathbf{S}({}^{p(j)}\mathbf{p}_j)^T & {}^j\mathbf{R}_{p(j)} \end{bmatrix} \quad (6)$$

where ${}^{p(j)}\mathbf{p}_j$ represents the position vector from coordinate frame $p(j)$ to the j coordinate frame, and $\mathbf{S}({}^{p(j)}\mathbf{p}_j)$ represents the transformation of the position vector ${}^{p(j)}\mathbf{p}_j$ into a skew-symmetric matrix.

The spatial momentum of the each link can be calculated from the spatial velocity as follow:

$$\mathbf{h}_j = \begin{bmatrix} \mathbf{k}_j \\ \mathbf{l}_j \end{bmatrix} = \mathbf{I}_j \mathbf{v}_j \tag{7}$$

where \mathbf{k}_j is the angular momentum, \mathbf{l}_j represents the linear momentum, and \mathbf{I}_j is the spatial inertia, all in reference to coordinate frame j .

Gathering all of the link velocities and joint velocities together, the system Jacobian \mathbf{J}_j can be defined to give the relationship between the two:

$$\mathbf{v}_j = \mathbf{J}_j \dot{\mathbf{q}} \tag{8}$$

The total momentum of a humanoid can be calculated by adding up all the angular and linear momentums contributed by the individual link segments, which can be described as follow:

$$\mathbf{h} = \sum_{j=1}^N {}^i \mathbf{X}_G^T \mathbf{h}_j = \sum_{j=1}^N {}^j \mathbf{X}_G^T \mathbf{I}_j \mathbf{J}_j \dot{\mathbf{q}} = \mathbf{A}_G \dot{\mathbf{q}} \tag{9}$$

where ${}^j \mathbf{X}_G$ denotes a transformation matrix that links the j coordinate from the centroidal frame, \mathbf{A}_G is the centroidal angular momentum matrix, and $\dot{\mathbf{q}}$ denotes the joint velocity vector.

The total momentum of a humanoid robot is described as follows:

$$\mathbf{h} = \mathbf{A}_G \dot{\mathbf{q}} \tag{10}$$

Consequently, the kinematics centroidal momentum is obtained. Next, the whole-body kinematic optimization problem is described in detail. The robot’s states are used to determine decision variables, including joint position \mathbf{q} , joint velocity $\dot{\mathbf{q}}$, and joint acceleration $\ddot{\mathbf{q}}$, which are employed as the control values. Here, we utilize the same knot points N , time durations T , and contact sequence \mathbf{c}_i . The whole-body kinematic optimization is formulated as follows:

$$\min_{\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}} \sum_{k=1}^N \left[\phi_k^{\text{kin}}(\cdot) + \Phi_k^{\text{kin}} \left(\begin{array}{c} \mathbf{h}[k] - {}^{\text{cen}} \mathbf{h}[k] \\ \mathbf{p}_i[k] - {}^{\text{cen}} \mathbf{p}_i[k] \end{array} \right) \right] \tag{11a}$$

$$\mathbf{h}[k] = \mathbf{A}_G(\mathbf{q}[k]) \dot{\mathbf{q}}[k] \tag{11b}$$

$$\dot{\mathbf{q}}[k+1] - \dot{\mathbf{q}}[k] = \ddot{\mathbf{q}}[k] \Delta t, \mathbf{q}[k+1] - \mathbf{q}[k] = \dot{\mathbf{q}}[k] \Delta t \tag{11c}$$

$$\dot{\mathbf{q}}_{\min} \leq \dot{\mathbf{q}}[k] \leq \dot{\mathbf{q}}_{\max}, \mathbf{q}_{\min} \leq \mathbf{q}[k] \leq \mathbf{q}_{\max} \tag{11d}$$

$$\dot{\mathbf{q}}_L[k] = \dot{\mathbf{q}}_R[k] \tag{11e}$$

$$\left(\mathbf{p}_i^{xy}[k] - \mathbf{p}_i^{xy}[k] \right) \mathbf{c}_i[k] = 0 \tag{11f}$$

$$\mathbf{p}_i^z[k] - h_{\text{terrain}} \geq 0 \tag{11g}$$

$$\ell^{\min} \leq \|\mathbf{p}[k] - \mathbf{r}[k]\|_2 \leq \ell^{\max} \tag{11h}$$

The quadratic cost (see Equation (11a)) includes a user-defined cost $\phi_k^{\text{kin}}(\cdot)$ and consensus cost Φ_k^{kin} . In detail, the user-defined cost $\phi_k^{\text{kin}}(\cdot)$ is used to minimize joint acceleration, which can be expressed as follows:

$$\phi_k^{\text{kin}}(\cdot) = \|\ddot{\mathbf{q}}[k]\|_{\mathbf{Q}_q}^2 \tag{12}$$

where $\mathbf{Q}_q \in \mathbb{R}^{n+6 \times n+6}$ denotes the weight matrix.

The consensus cost Φ_k^{kin} penalizes deviations in the CoM position ${}^{\text{cen}} \mathbf{r}$, centroidal momentum ${}^{\text{cen}} \mathbf{h}$, and foot positions ${}^{\text{cen}} \mathbf{p}_i$ from the solution of the centroidal momentum,

which enforces consistency between the dynamic and whole-body kinematic problems. It can be set as follows:

$$\Phi_k^{\text{kin}}(\cdot) = \|\mathbf{h}[k] - \text{cen}\mathbf{h}[k]\|_{\mathbf{Q}_{fh}}^2 + \|\mathbf{r}[k] - \text{cen}\mathbf{r}[k]\|_{\mathbf{Q}_{fr}}^2 + \|\mathbf{p}_i[k] - \text{cen}\mathbf{p}_i[k]\|_{\mathbf{Q}_{fp}}^2 \quad (13)$$

where \mathbf{h} , \mathbf{r} , and \mathbf{p}_i are calculated by the floating base kinematics, and $\mathbf{Q}_{fh} \in \mathbb{R}^{6 \times 6}$, $\mathbf{Q}_{fr} \in \mathbb{R}^{3 \times 3}$, and $\mathbf{Q}_{fp} \in \mathbb{R}^{3 \times 3}$ are weight matrices.

Integration constraint: The constraints of the whole-body kinematic include the dynamical constraint (see Equation (11b)) and integration constraint (see Equation (11c)).

Joint constraint: The joint position should be limited within \mathbf{q}_{\min} and \mathbf{q}_{\max} to avoid exceeding the joint position limits. Similarly, joint velocities should also be restricted (Equation (11d)). More importantly, \mathbf{q}_L represents the joint positions of the left leg and the left arm, and \mathbf{q}_R represents the joint positions of the right leg and the right arm. The joint velocities of $\dot{\mathbf{q}}_L$ and $\dot{\mathbf{q}}_R$ should be consistent because of the symmetry of the robot (Equation (11e)).

Terrain constraint: The tangential displacement between the two knot points must be zero when the foot does not contact the ground. This constraint disappears during the flight phase. Equation (11f,g) guarantee that the foot position remains above the terrain surface.

Distance constraint: The distance constraint is expressed in Equation (11h). By tracking the reference variables provided by the centroidal dynamics optimization, a high vertical jumping trajectory can be obtained through whole-body kinematic optimization.

3. Simulation and Experimental Results

3.1. Validation Setup

The humanoid robot platform, model parameters, and solution for the optimization problem are described below.

The proposed optimization method was demonstrated on a humanoid robot, as shown in Figure 2. This humanoid robot has a total mass of 42.6 kg, a height of 1.5 m, and 14 degrees of freedom. The hip and arm joints are driven by a quasi-drive actuator composed of a brushless DC motor and a harmonic reducer, whereas the knee and ankle joints are driven by a brushless DC motor with ball screw structures. These joints can smoothly and instantaneously switch between position mode and torque mode [17]. There are two six-axis force/torque sensors (M3714B2, Sunrise Instruments, Canton, MI, USA) that measure contact force/torque. Furthermore, a Modbus communication system is established using EtherCAT, with a control cycle time of 1 ms. The model parameters of the humanoid robot are derived from the robot CAD model. Considering the large number of robot connecting rods, the masses and inertias of these links are not listed separately.

The FMINCON function in MATLAB is used to construct and solve the single sequential kino-dynamic trajectory optimization problem for jumping behaviors [36]. This tool allows for flexibility in defining system dynamics, path constraints, and boundary constraints as per user requirements. Additionally, the optimization program is executed on a quadcore computer (Intel Core i7-7700, 3.60 GHz).

3.2. Numerical Optimization

Using the single sequential kino-dynamic trajectory optimization, a vertical jumping trajectory with a vertical height of 0.5 m (foot lifting distance) was generated that includes both the launching and flight phases. The discrepancies in consistency between the centroidal dynamics and whole-body kinematics, especially regarding momentum and foot position, are discussed in detail. In addition, the main parameters used in the optimization process can be found in Table 1.

Table 1. Main parameters for the single sequential kino-dynamic trajectory optimization.

Main Parameter	Value
Knot points, N	20
Motion duration, T	1 s
Contact sequence of c_1, c_2, \dots and c_8	[1, ..., 1, 0, 0, 0, 0, 0, 0]
Optimality tolerance	$<1 \times 10^{-5}$
Q_{cf}	diag (0.001, 0.001, 0.001, ..., 0.001, 0.001, 0.001)
Q_{cr}	diag (40, 40, 40)
Q_{ch}	diag (30, 10, 10, 10, 50, 10)
Q_{ce}	diag (0.01, 0.01, 0.01, ..., 0.01, 0.01, 0.01)
Q_q	diag (0.0001, 0.0001, 0.0001, ..., 0.0001)
Q_{fh}	diag (1, 1, 1, 1, 1)
Q_{fr}	diag (0.01, 0.01, 0.01)
Q_{fp}	diag (0.02, 0.02, 0.02, ..., 0.02, 0.02, 0.02)
Q_{fe}	diag (0.01, 0, 0.001, ..., 0.01, 0, 0.001)

Table 2 lists the main parameters for the numerical optimization, such as the number of variables and the number of equality constraints, and also presents the solving times for both optimization problems. The solving times for centroidal dynamics and whole-body kinematics optimization are 0.230 s and 3.2 s, respectively. In addition, the trend of the cost value with iterations is shown in Figure 3. As the number of iterations increases, the cost value tends to be stable, indicating that the optimization problem is approaching convergence. Therefore, it is evident that this optimization process is highly efficient and takes little time.

Table 2. Numerical optimization of main parameters.

Optimization Project	Centroidal Dynamics	Whole-Body Kinematics
Number of variables	720	1200
Duration of motion	1 s	1 s
Number of equality constraints	1036	720
Number of inequality constraints	3102	4250
Solving time	0.230 s	3.2 s
Number of iterations	12	35

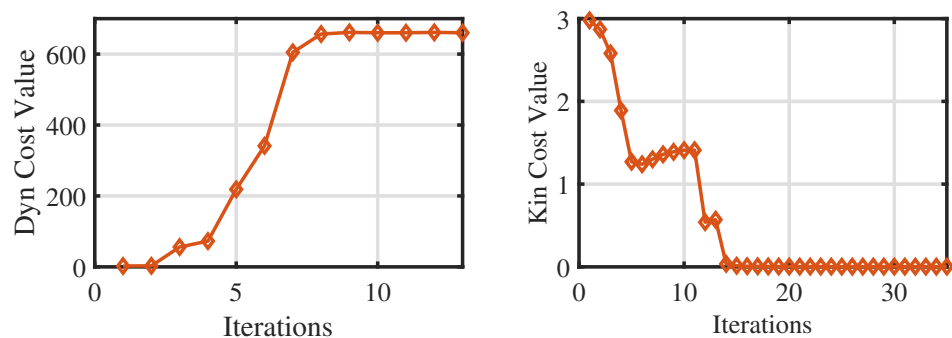


Figure 3. The trend of the cost value of the centroidal dynamics and whole-body kinematics with iterations.

From Figures 4 and 5, the consistency of the momentum and CoM at each knot is high. Ensuring the consistency of momentum and CoM is key when bridging the centroidal dynamic and whole-body kinematic optimization, and hence, the relevant weight matrices Q_{fr} and Q_{fh} in Equation (13) are adjusted to maintain momentum consistency. Notably, the consistency of the foot position is poor (see Figure 6), and even the motion trends in

the x -direction are quite different. This observed phenomenon can be attributed to the centroidal dynamics model, which focuses primarily on the relative position between the CoM and the foot position without introducing whole-body kinematic constraints. Because the optimized foot position might not meet the whole-body kinematic constraints, it is treated merely as a reference trajectory during the whole-body optimization, and high consistency is not pursued.

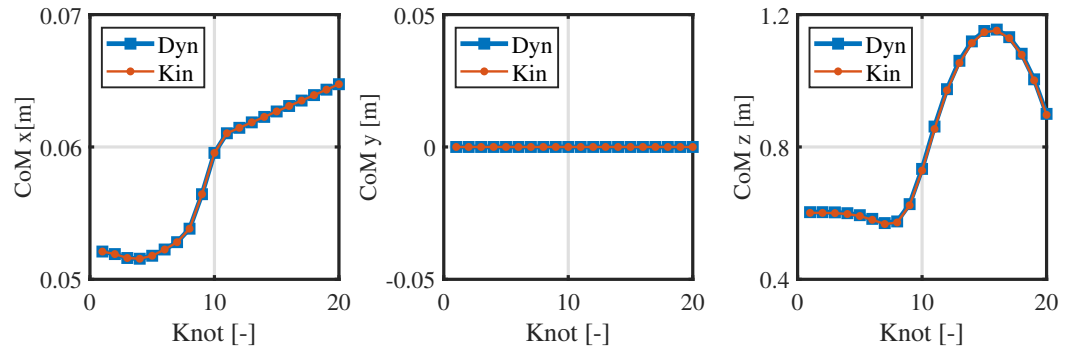


Figure 4. Consistency in the CoM of the centroidal dynamics and whole-body kinematics.

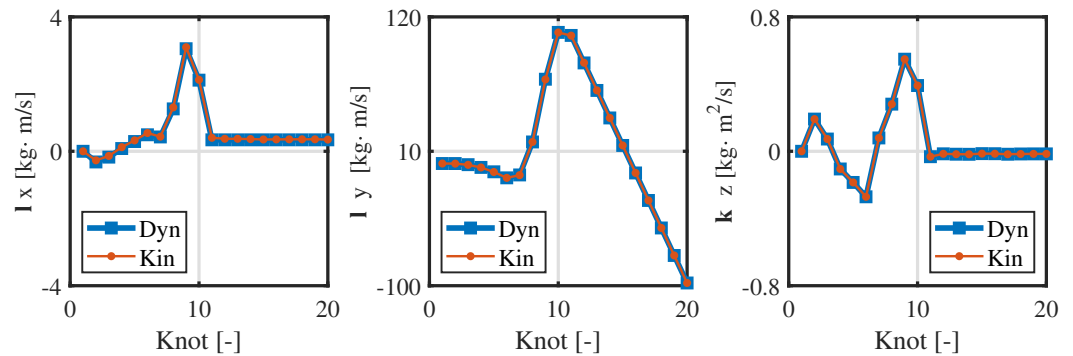


Figure 5. Consistency in the momentum of the centroidal dynamics and whole-body kinematics.

Figures 6 and 7 presents the positions and contact forces at the contact point c_1 . When this point is touching the ground, there is obviously no position sliding and the normal contact force exists; during the flight phase, the contact force is zero and the position changes. These fully meet the complementarity conditions (see Equations (2h,i) and (11f)). This result verifies the effectiveness of the complementary conditions and its combination with the re-specified contact sequence. Finally, the whole-body trajectory including the launching and flight phases is obtained.

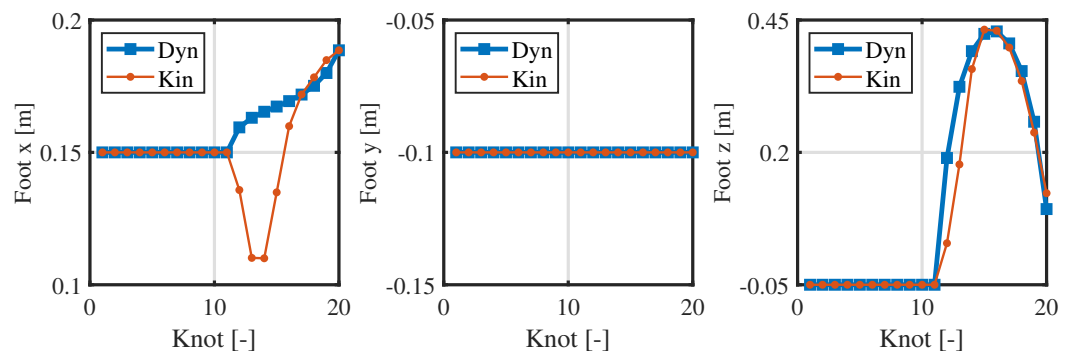


Figure 6. Consistency in the foot position of the centroidal dynamics and whole-body kinematics at the contact point c_1 .

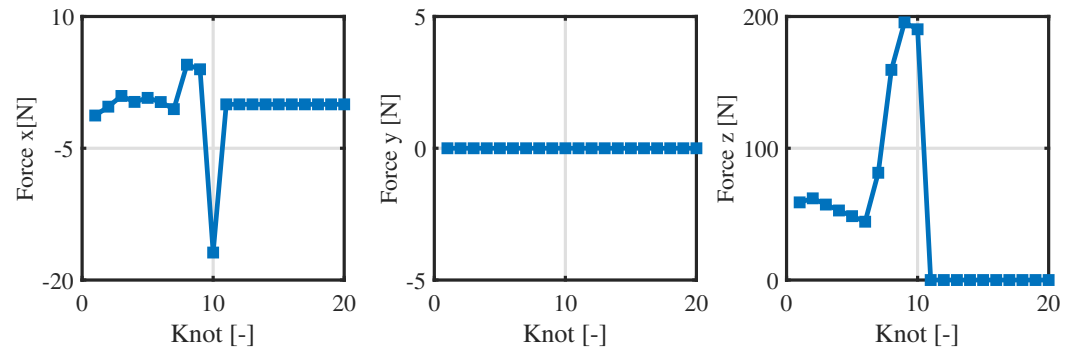


Figure 7. Optimized contact forces of contact point c_1 in the centroidal dynamics optimization.

To sum up, the whole-body trajectory of the vertical jump can be optimized using a single sequential optimization, and it is an efficient solution while saving time. After solving the optimization problem, the state values and control values that satisfy the constraints of the centroidal dynamics and whole-body kinematics are obtained. Subsequently, the whole-body trajectory is derived by utilizing a spline function to interpolate the states \mathbf{q} between two knot points, which were used for the simulation and experiment (see Figure 13).

3.3. Simulations

The effectiveness of the optimized trajectory was verified using CoppeliaSim dynamics software. During the launching and flight phases, the whole-body trajectory was performed in the joint position mode of the robot. Then, the joints smoothly switched from position mode to torque mode. For the landing phase, a landing controller that had been designed for our team's previous research [17] was adopted. Here, the CoM trajectory was derived from the centroidal dynamics optimization.

Figure 8 shows a snapshot of the jumping simulation. The humanoid robot was able to achieve a stable launch as the planned jumping motion. Figure 9 illustrates the tracking of the CoM position and angular momentum in the launching phase. Obviously, the CoM tracks well in the z and x directions, and the maximum tracking error is less than 3 mm when taking off in the x direction, which is well within tolerance. In addition, the angular momentum around the y -axis also has a good tracking effect, and the angular momentum is close to the reference value at the moment of take-off.

Additionally, the robot has an obvious leg retraction motion in the first half of the flight phase, which results in a high jump (foot lifting distance). Figure 10 depicts the absolute positions of the CoM and foot as well as their relative distance. The CoM rises by 0.30 m, and the foot retracts concurrently by 0.18 m. This coordinated motion enables the robot to achieve a jump height of 0.48 m from the ground. According to Equation (4) in Section 2.2, the relative distance between the foot and CoM can be adjusted by changing the weight matrix \mathbf{Q}_{ce} . In addition, the arm's motion can be leveraged more effectively by adjusting the weight matrix \mathbf{Q}_q , which prevents excessive trunk bending and ensures a stable posture during both the launching and flight phases.

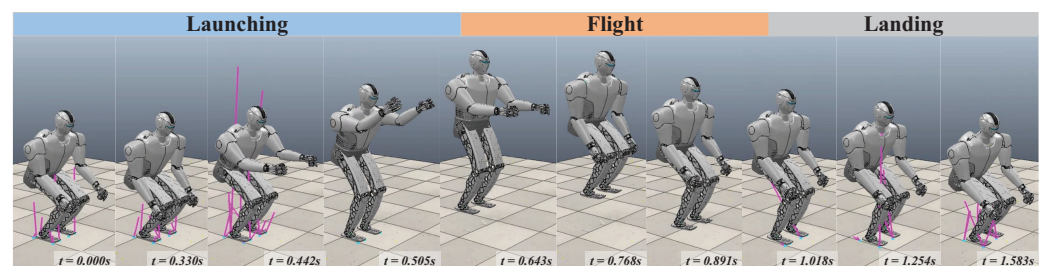


Figure 8. Snapshot of the simulation in the CoppeliaSim simulator with the optimized trajectory (Supplementary Materials).

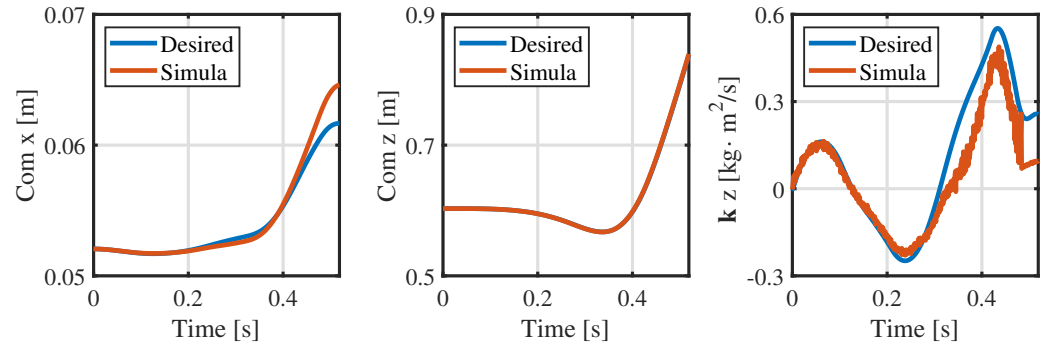


Figure 9. CoM position and angular momentum during the launching phase in the simulation.

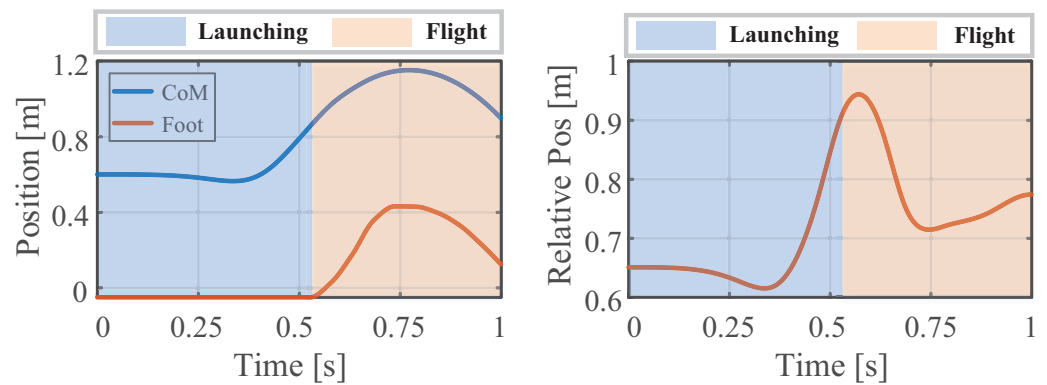


Figure 10. Absolute and relative positions of the humanoid robot’s CoM and foot under vertical jump conditions.

3.4. Experiments

The same whole-body trajectory used in the simulation was also executed on a real humanoid robot, as depicted in Figure 11. Clearly, the robot can achieve stable launching similar to the simulation, then retract its legs during the flight phase to achieve a 0.5 m (foot lifting distance) vertical jump, and subsequently maintain a stable posture until landing.

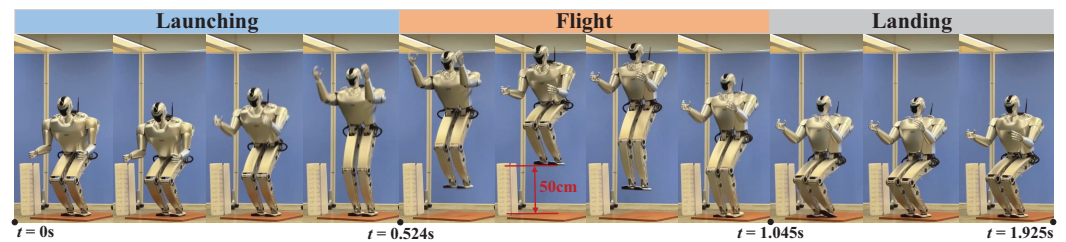


Figure 11. Snapshot of the experiment with a 0.5 m (foot lifting distance) vertical jump (Supplementary Materials).

Figure 12 illustrates tracking of the CoM position and momentum in the launching phase. The joint angle and velocity are filtered by a low-pass filter, and the CoM position and momentum data are effectively computed through the forward kinematic and momentum equation. The actual CoM position tracks well in the z direction. Moreover, the tracking effectiveness in the x direction is not satisfactory, but the general trend is roughly the same. More importantly, the actual momentum around the y -axis is tracked satisfactorily in the initial 0.4 s, and then the actual angular momentum rises rapidly to about $2.5 \text{ kg} \cdot \text{m}^2/\text{s}$ and then rises sharply to $0.2 \text{ kg} \cdot \text{m}^2/\text{s}$ at the moment of take-off.

The reason for this phenomenon may be poor tracking of the joint angle and velocity, particularly in the late launching phase. Taking the joint motion of the robot’s left part as an example, Figures 13 and 14 illustrate, respectively, the tracking performance of the joint

angle and velocity in the launching and flight phases. There are certain tracking errors in the angles and velocities of the hip, knee, and ankle joints, coupled with the fact that the actual contact force in the x -direction is much larger than desired (the desired contact forces at the contact surface vertices are converted into a six-dimensional force at the footplate [33], and the results are shown in Figure 15), which together make the robot move forward in the x -direction. As a result, the pressure center of the foot gradually moves toward the toes, causing the heel to lift. The heel is raised about 2~3 cm in height, as can be seen in Figure 11. This may lead to fluctuation in the contact force and contact torque at about s , which, in turn, causes poor tracking of the angular momentum.

Figures 13 and 14 also compare the optimized whole-body trajectory with those of our previous research [17]. (The joint trajectories collected in the dynamic simulation are combined with reaction momentum control (RMC) [37] to generate the whole-body trajectory for the launching and flight phases. All jump heights are 50 cm, and the joint trajectories are represented by the icon labeled 'TRO'.) The maximum speeds of the hips, knees, and ankles decrease by varying degrees, and the velocity curve is smoother. Therefore, the proposed optimization method not only generates more usable trajectories but also obtains them more quickly.

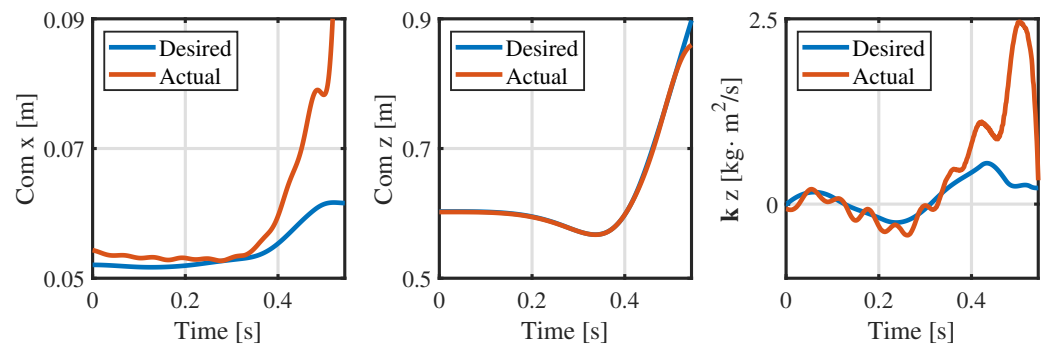


Figure 12. CoM position and angular momentum in the launching phase in the experiment.

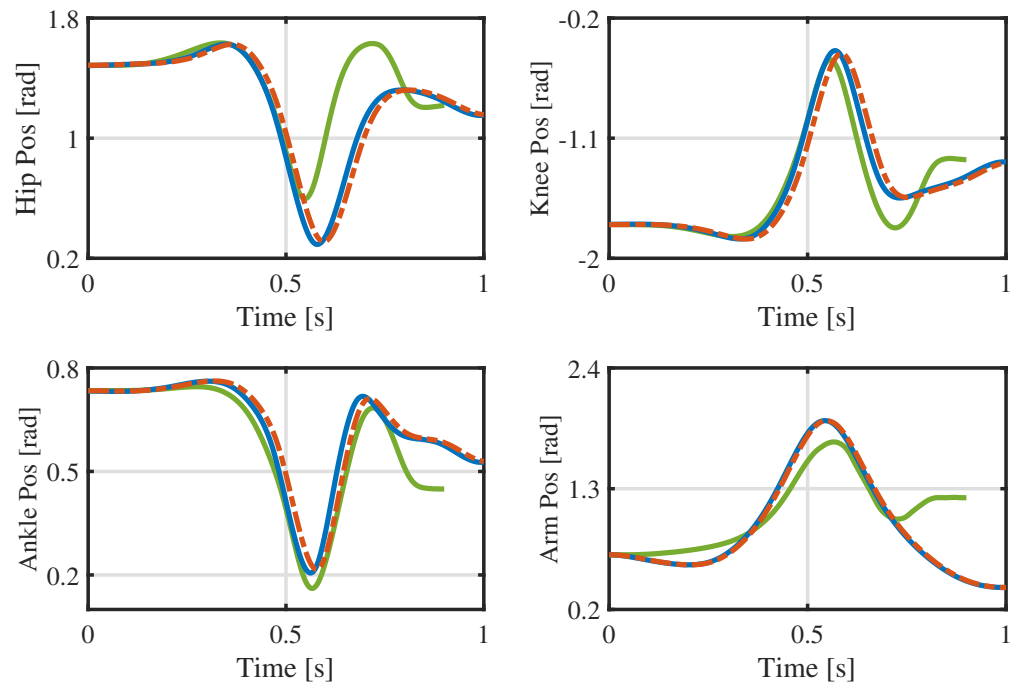


Figure 13. Joint angles in the launching and flight phases of the experiment.

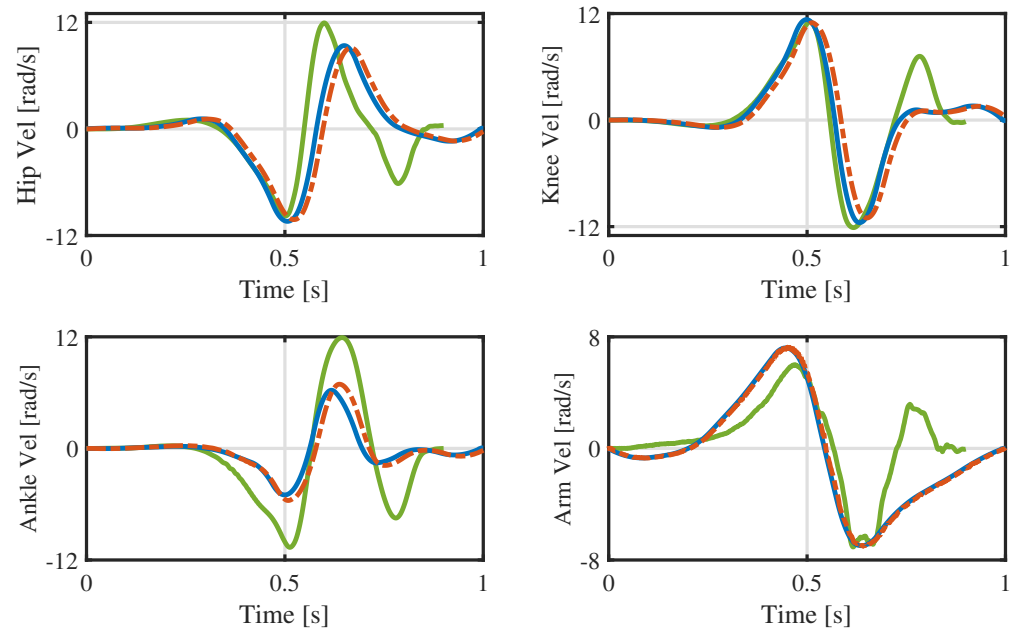


Figure 14. Joint velocities in the launching and flight phases of the experiment.

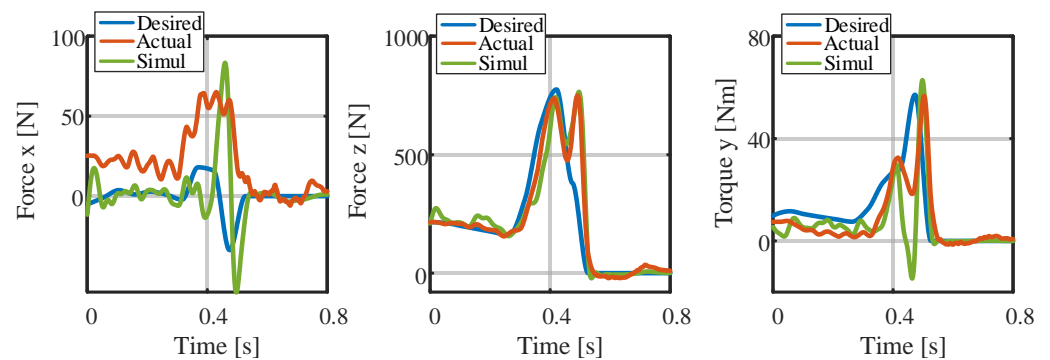


Figure 15. Contact forces in the launching and flight phases of the experiment.

The experimental results show that the optimized trajectory successfully realizes the desired vertical jumping motion during the launching and flight phases, and hence, the effectiveness of the proposed method has been verified.

4. Conclusions

High vertical jumping motion directly reflects a humanoid robot's extreme motion capabilities and enables it to leap over obstacles. This article proposed a novel sequential dynamic trajectory optimization method that sequentially optimizes the centroidal dynamics and whole-body kinematics to solve the whole-body trajectory for high vertical jump motions. The proposed method is an efficient solution strategy that significantly reduces the solving time: giving a total optimization time of approximately 3.43 s. To validate the effectiveness of the single sequential kino-dynamic trajectory optimization, the optimized whole-body trajectory was verified through simulation and experimentation on a humanoid robot platform. The results show that the CoM position and momentum during the launching phase are satisfactorily tracked; the humanoid robot can achieve a stable vertical jump of 0.5 m. After comparing the optimized trajectory with the trajectory collected in the dynamic simulation and RMC, we found that the maximum joint velocity decreases by varying degrees, demonstrating that the optimized trajectory is more suitable for jump motion planning. The proposed method can be applied to a bipedal robot, and it is suitable for humanoid robots, which have high degrees of freedom. In the future, the

proposed method will be enhanced to achieve online whole-body trajectory optimization and provide more practical whole-body trajectories combined with environment-aware information. Additionally, we plan to achieve more agile kinds of jump motions, such as a back flip.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/biomimetics9050274/s1>.

Author Contributions: Conceptualization, Y.L.; methodology, Y.L. and H.Q.; software, C.Y.; validation, Y.L., H.Q. and C.Y.; formal analysis, Y.L. and H.Q.; investigation, Y.L.; writing—original draft preparation, Y.L.; writing—review and editing, Y.L.; visualization, X.C.; supervision, X.C.; project administration, X.C. and Z.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under grant (62073041) and in part by the Discipline Innovation and Talent Introduction Program in Colleges and Universities (111 Program) (B08043).

Institutional Review Board Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Choi, B.; Lee, W.; Park, G.; Lee, Y.; Min, J.; Hong, S. Development and control of a military rescue robot for casualty extraction task. *J. Field Robot.* **2019**, *36*, 656–676. [[CrossRef](#)]
2. Kumagai, I.; Morisawa, M.; Sakaguchi, T.; Nakaoka, S.; Kaneko, K.; Kaminaga, H.; Kajita, S.; Benallegue, M.; Cisneros, R.; Kanehiro, F. Toward industrialization of humanoid robots: Autonomous plasterboard installation to improve safety and efficiency. *IEEE Robot. Autom. Mag.* **2019**, *26*, 20–29. [[CrossRef](#)]
3. Yoshiike, T.; Kuroda, M.; Ujino, R.; Kanemoto, Y.; Kaneko, H.; Higuchi, H.; Komura, S.; Iwasaki, S.; Asatani, M.; Koshiishi, T. The experimental humanoid robot e2-dr: A design for inspection and disaster response in industrial environments. *IEEE Robot. Autom. Mag.* **2019**, *26*, 46–58. [[CrossRef](#)]
4. Cai, Z.; Zhu, X.; Gergondet, P.; Chen, X.; Yu, Z. A Friction-Driven Strategy for Agile Steering Wheel Manipulation by Humanoid Robots. *Cyborg Bionic Syst.* **2023**, *4*, 0064. [[CrossRef](#)] [[PubMed](#)]
5. Wensing, P.M.; Orin, D.E. Generation of dynamic humanoid behaviors through task-space control with conic optimization. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 3103–3109.
6. Ahn, D.; Cho, B.K. Optimal standing jump trajectory generation for biped robots. *Int. J. Precis. Eng. Manuf.* **2020**, *21*, 1459–1467. [[CrossRef](#)]
7. Vatavuk, I.; Kovačić, Z. Precise Jump Planning using Centroidal Dynamics based Bilevel Optimization. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 3026–3032.
8. Zhu, Q.; Wu, W.; Zhao, Y.; Wu, J.; Xiong, R. Design and control of stiffness coordination between knee and ankle joints for a hopping legged robot. In Proceedings of the 2017 IEEE International Conference on Unmanned Systems (ICUS), Beijing, China, 27–29 October 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 624–630.
9. Wensing, P.M.; Orin, D.E. Improved computation of the humanoid centroidal dynamics and application for whole-body control. *Int. J. Humanoid Robot.* **2016**, *13*, 1550039. [[CrossRef](#)]
10. Chignoli, M.; Kim, D.; Stanger-Jones, E.; Kim, S. The MIT humanoid robot: Design, motion planning, and control for acrobatic behaviors. In Proceedings of the 2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids), Munich, Germany, 19–21 July 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–8.
11. Wan, X.; Urakubo, T.; Tada, Y. Optimization of jumping motion of a legged robot for different take-off postures. *J. Mech. Sci. Technol.* **2015**, *29*, 1391–1397. [[CrossRef](#)]
12. Xiong, X.; Ames, A.D. Sequential motion planning for bipedal somersault via flywheel slip and momentum transmission with task space control. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 25–29 October 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 3510–3517.
13. Dynamics, B. What's New, Atlas? 2017. Available online: <https://youtube.com/watch?v=fRj34o4hN4I/> (accessed on 27 April 2024).

14. Kojima, K.; Kojio, Y.; Ishikawa, T.; Sugai, F.; Kakiuchi, Y.; Okada, K.; Inaba, M. A robot design method for weight saving aimed at dynamic motions: Design of humanoid JAXON3-P and realization of jump motions. In Proceedings of the 2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids), Toronto, ON, Canada, 15–17 October 2019; IEEE: Piscataville, NJ, USA, 2019; pp. 586–593.
15. Ahn, D.; Cho, B.K. Online Jumping Motion Generation via Model Predictive Control. *IEEE Trans. Ind. Electron.* **2021**, *69*, 4957–4965. [[CrossRef](#)]
16. Xiong, X.; Ames, A.D. Bipedal hopping: Reduced-order model embedding via optimization-based control. In Proceedings of the 2018 IEEE/RISJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; IEEE: Piscataville, NJ, USA, 2018; pp. 3821–3828.
17. Qi, H.; Chen, X.; Yu, Z.; Huang, G.; Liu, Y.; Meng, L.; Huang, Q. Vertical Jump of a Humanoid Robot with CoP-Guided Angular Momentum Control and Impact Absorption. *IEEE Trans. Robot.* **2023**, *39*, 3154–3166. [[CrossRef](#)]
18. Zhang, J.; Shen, J.; Liu, Y.; Hong, D. Design of a jumping control framework with heuristic landing for bipedal robots. In Proceedings of the 2023 IEEE/RISJ International Conference on Intelligent Robots and Systems (IROS), Detroit, MI, USA, 1–5 October 2023; IEEE: Piscataville, NJ, USA, 2023; pp. 8502–8509.
19. Xiang, P.; Yan, L.; Guo, Y.; He, X.; Gerada, C.; Chen, I.M. A concentrated-flux-type pm machine with irregular magnets and iron poles. *IEEE/ASME Trans. Mechatron.* **2023**, *29*, 691–702. [[CrossRef](#)]
20. Zhou, Z.; Wingo, B.; Boyd, N.; Hutchinson, S.; Zhao, Y. Momentum-aware trajectory optimization and control for agile quadrupedal locomotion. *IEEE Robot. Autom. Lett.* **2022**, *7*, 7755–7762. [[CrossRef](#)]
21. Chen, H.; Wang, B.; Hong, Z.; Shen, C.; Wensing, P.M.; Zhang, W. Underactuated motion planning and control for jumping with wheeled-bipedal robots. *IEEE Robot. Autom. Lett.* **2020**, *6*, 747–754. [[CrossRef](#)]
22. Bledt, G.; Kim, S. Implementing regularized predictive control for simultaneous real-time footstep and ground reaction force optimization. In Proceedings of the 2019 IEEE/RISJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; IEEE: Piscataville, NJ, USA, 2019; pp. 6316–6323.
23. Jeon, S.H.; Kim, S.; Kim, D. Online Optimal Landing Control of the MIT Mini Cheetah. In Proceedings of the 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23–27 May 2022; IEEE: Piscataville, NJ, USA, 2022; pp. 178–184.
24. Winkler, A.W.; Bellicoso, C.D.; Hutter, M.; Buchli, J. Gait and trajectory optimization for legged systems through phase-based end-effector parameterization. *IEEE Robot. Autom. Lett.* **2018**, *3*, 1560–1567. [[CrossRef](#)]
25. Dai, H.; Valenzuela, A.; Tedrake, R. Whole-body motion planning with centroidal dynamics and full kinematics. In Proceedings of the 2014 IEEE-RAS International Conference on Humanoid Robots, Madrid, Spain, 18–20 November 2014; IEEE: Piscataville, NJ, USA, 2014; pp. 295–302.
26. Ponton, B.; Khadiv, M.; Meduri, A.; Righetti, L. Efficient multicontact pattern generation with sequential convex approximations of the centroidal dynamics. *IEEE Trans. Robot.* **2021**, *37*, 1661–1679. [[CrossRef](#)]
27. Budhiraja, R.; Carpentier, J.; Mansard, N. Dynamics consensus between centroidal and whole-body models for locomotion of legged robots. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; IEEE: Piscataville, NJ, USA, 2019; pp. 6727–6733.
28. Nguyen, C.; Nguyen, Q. Contact-timing and trajectory optimization for 3d jumping on quadruped robots. In Proceedings of the 2022 IEEE/RISJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 23–27 October 2022; IEEE: Piscataville, NJ, USA, 2022; pp. 11994–11999.
29. Neunert, M.; Farshidian, F.; Winkler, A.W.; Buchli, J. Trajectory optimization through contacts and automatic gait discovery for quadrupeds. *IEEE Robot. Autom. Lett.* **2017**, *2*, 1502–1509. [[CrossRef](#)]
30. Deits, R.; Tedrake, R. Footstep planning on uneven terrain with mixed-integer convex optimization. In Proceedings of the 2014 IEEE-RAS International Conference on Humanoid Robots, Madrid, Spain, 18–20 November 2014; IEEE: Piscataville, NJ, USA, 2014; pp. 279–286.
31. Aceituno-Cabezas, B.; Mastalli, C.; Dai, H.; Focchi, M.; Radulescu, A.; Caldwell, D.G.; Cappelletto, J.; Grieco, J.C.; Fernández-López, G.; Semini, C. Simultaneous contact, gait, and motion planning for robust multilegged locomotion via mixed-integer convex optimization. *IEEE Robot. Autom. Lett.* **2017**, *3*, 2531–2538.
32. Herzog, A.; Schaal, S.; Righetti, L. Structured contact force optimization for kino-dynamic motion generation. In Proceedings of the 2016 IEEE/RISJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Republic of Korea, 9–14 October 2016; IEEE: Piscataville, NJ, USA, 2016; pp. 2703–2710.
33. Caron, S.; Pham, Q.C.; Nakamura, Y. Stability of surface contacts for humanoid robots: Closed-form formulae of the contact wrench cone for rectangular support areas. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; IEEE: Piscataville, NJ, USA, 2015; pp. 5107–5112.
34. Caron, S.; Kheddar, A. Multi-contact walking pattern generation based on model preview control of 3D COM accelerations. In Proceedings of the 2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids), Cancun, Mexico, 15–17 November 2016; IEEE: Piscataville, NJ, USA, 2016; pp. 550–557.
35. Chignoli, M.; Kim, S. Online trajectory optimization for dynamic aerial motions of a quadruped robot. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; IEEE: Piscataville, NJ, USA, 2021; pp. 7693–7699.

-
36. Kelly, M. An Introduction to Trajectory Optimization: How to Do Your Own Direct Collocation. *SIAM Rev.* **2017**, *59*, 849–904. [[CrossRef](#)]
 37. Lee, S.H.; Goswami, A. A momentum-based balance controller for humanoid robots on non-level and non-stationary ground. *Auton. Robot.* **2012**, *33*, 399–414. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.