

## Research Article

# Discretized Mid-Value CLVI-PDNN Based Redundancy Resolution for Single Leg of Quadruped Robot

Jinhui Rao <sup>1</sup>, Taihui Zhang <sup>1,2</sup>, Honglei An,<sup>1</sup> and Hongxu Ma<sup>1</sup>

<sup>1</sup>College of Intelligence Science and Technology, National University of Defense Technology, Changsha 410073, Hunan Province, China

<sup>2</sup>Department of Aerospace Medicine, The Air Force Medical University, Xi'an 710000, Shanxi Province, China

Correspondence should be addressed to Taihui Zhang; zhangtaihui@126.com

Received 12 April 2019; Accepted 19 May 2019; Published 24 June 2019

Academic Editor: Oscar Reinoso

Copyright © 2019 Jinhui Rao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The two most important performance indicators of quadruped robot are load capacity and walking speed, and these performance indicators of the whole robot finally reflect on the joint torques and angular velocities. To satisfy different requirements of walking speed and load capacity when quadruped robots implement different tasks, the joint torques and angular velocities need to be balanced with physical constraints of the joints. A single leg with redundant DOF (degree of freedom) could optimize the distribution of joint torques or angular velocities based on different performance requirements. This paper presents a kind of new recurrent neural networks taking joint torques and angular velocities simultaneously into consideration and proposes mid-value CLVI-PDNN to achieve the optimal joint torques and angular velocities with physical constraints of the mechanism as described in our previous paper. Because the continuous mid-value CLVI-PDNN has difficulty in real-time operation because of too much calculation workload, two kinds of methods are proposed to discretize the mid-value CLVI-PDNN for application on computer or digital circuit. The simulation results demonstrate the efficacy of the algorithm proposed in this paper.

## 1. Introduction

Legged robots could be applied for rescue or service in unstructured environment due to its agility on rough terrain. Because of much more motion stability than biped robots and less complexity compared to hexapod robots or octopod robots, quadruped robots have a broad application prospect. Figure 1 shows the hydraulic actuated quadruped robot developed by our laboratory, which has four legs assembled symmetrically, four revolute joints on each leg, a torso with computer, and other electronic equipment on board.

The schematic diagram of the single leg is shown in Figure 2, which has a redundant joint in the sagittal plane. The moment arms of actuating forces are marked in Figure 2 using red lines. If the performance of the hydraulic actuator is finitude, the maximum joint torques and angular velocity are determined by the length of these red lines. Dimension design of the moment arms of actuating forces directly affects the performance of walking speed and load capacity of the robot.

Designing the mechanism dimensions of the legs especially the moment arms needs to optimize redundancy

resolution of the single leg depending on performance demands then distribute joint torques and angular velocity on the condition of joint physical constraints. Simultaneously, optimizing the distribution of joint torques and angular velocity could be used for designing the control algorithm of the robot. Generally, two kinds of methods are widely used to deal with inverse kinematics of redundant robots, i.e., analytical method and numerical analysis.

In some analytical methods, inverse kinematics of redundant robot could be described as a parameterized function with parameters including joint angles [1, 2], the angle between the reference plane and robot [3, 4]. Manually adding constraints is another analytical method to calculate the inverse kinematics of redundant robot [5]. The inverse kinematics problem could be analytically solved while the redundant DOFs matches number of added constraints.

Because redundant robot systems are generally strongly nonlinear and highly coupled, it is not a simple task to resolve inverse kinematics problem. Redundancy resolution is not only beneficial to obstacles avoidance and mechanism physical constraints avoidance [6–8] but also is used for

improving motion performance based on diverse criteria [9–12]. Pseudoinverse type solution is a kind of widely used redundancy resolution [13, 14]. Some researchers solved inverse kinematics problem using different methods including interpolation [15, 16] and gradient projection [17–19], but these methods were restricted to computation speed and error. A number of novel or so called intelligent numerical resolving methods are used for redundant inverse kinematics problem, including quadratic programming (QP), artificial neural networks, quaternion method, online learning algorithm, and genetic algorithm [20–24]. Among these methods, quadratic programming is tractable and has well expansibility to real-time application. Minimizing sum of squares of joint velocity is the most widely utilized optimization criterion which is computationally tractable but ignores dynamics at all [25]. Besides, the optimization criterion that minimizes sum of squares of joint torques is also exploited to resolve redundant inverse kinematics which may lead to instability.

To acquire a comprehensive performance combining kinematics and dynamics, Ma proposed an optimization criterion simultaneously minimizing joint velocity and torques [26]. Yunong Zhang also proposed an optimization criterion that consists of minimizing sum of squares of joint velocity and torques simultaneously [27]. The optimization problem on different level (i.e., joint velocity and joint torques) was converted into a standard QP problem with equality and inequality constraints on joint-acceleration level. Differing from pseudoinverse type solution used by Ma, Zhang numerically solved the constrained QP problem using a kind of recurrent neural network which is named linear variational inequality-based primal-dual neural network (LVI-PDNN) [28–30].

It is different from manipulators that legged robots alter between support phase and flight (swing) phase when they are walking or running. Therefore, the legs need to swing forward and support body in different phases alternatively. Accordingly, the legs need to provide sufficient joint velocity and joint torques in different phases alternatively. On the basis of the above analysis, a new kind of optimization criterion consisting of support phase and flight (swing) phase is proposed in our previous paper and converted onto angular acceleration level based on ZNN. A feasible distribution of the joint torques and angular velocity is obtained according to continuous mid-value CLVI-PDNN, but the algorithm used in our previous paper has a drawback, i.e., hard to real-time application because of too large computation cost [31].

For applying recurrent neural networks based real-time optimization on computer or digital circuit, discrete neural networks gradually attract attention from researchers. M.J. Perez-Illarbe proposed a kind of discrete recurrent neural network to solve bounded strictly convex quadratic programming problem [32]. Then he applied Wolf duality theory to form a kind of simple discrete recurrent neural network for solving linear constraints quadratic programming problem [33]. H. Tang et al. proposed a general recurrent neural network to solve hybrid constraints quadratic programming problem [34]. Q. Liu et al. used discrete recurrent neural network to solve equality constraints quadratic programming problem and proved the global exponential stability of its



FIGURE 1: Hydraulic actuated quadruped robot [31].

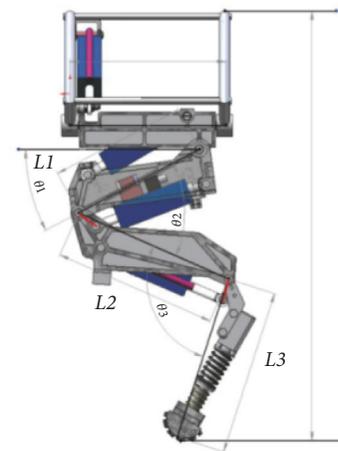


FIGURE 2: Schematic diagram of the single leg [31].

solution [35]. Y. Zhang, B. Liao, L. Jin et al. researched on a series of problems such as time-variant matrix inversion, time-variant polynomial root searching, and time-variant nonlinear optimization. They compared several discretization methods and proved the convergency of discrete recurrent neural network [36–43].

The main contributions of this paper are as follows: firstly, two kinds of discretization methods are proposed to discretize the mid-value CLVI-PDNN for computer control of the quadruped robot, i.e., bilinear transform-type and Taylor-type discretization methods, while the recursion formulas of the two kinds of discrete mid-value CLVI-PDNN are also presented; secondly, inherent nature of the two kinds of discretization algorithms is analyzed theoretically, including the order of truncation error and stability; thirdly, simulation results of the proposed methods indicate the efficacy for redundancy resolution of single leg.

## 2. The Optimization Criterion and QP Formulation

In this paper, the redundant single leg of the quadruped robot is illustrated to show how to deal with its inverse kinematics

problem using optimization method. Firstly, the kinematics and dynamics of the single leg system in sagittal plane with three joints are presented in this section. The kinematics of single leg is as follows:

$$r = f(\theta), \quad (1)$$

where  $\theta = [\theta_1 \ \theta_2 \ \theta_3]^T$  is the position of joints in joint space, while  $r = [r_1 \ r_2]^T$  is the position of foot in Cartesian space. The differential kinematics is as follows:

$$\dot{r} = J(\theta)\dot{\theta}, \quad (2)$$

where  $\dot{\theta} = [\dot{\theta}_1 \ \dot{\theta}_2 \ \dot{\theta}_3]^T$  is the velocity of joints in joint space, while  $\dot{r} = [\dot{r}_1 \ \dot{r}_2]^T$  is the velocity of foot in Cartesian space.  $J = \partial f / \partial \theta$  is Jacobian matrix. Second order differential equation is

$$\ddot{r} = J(\theta)\ddot{\theta} + \dot{J}(\theta)\dot{\theta}, \quad (3)$$

where  $\ddot{\theta} = [\ddot{\theta}_1 \ \ddot{\theta}_2 \ \ddot{\theta}_3]^T$  is the acceleration of joint in joint space, while  $\ddot{r} = [\ddot{r}_1 \ \ddot{r}_2]^T$  is the acceleration of foot in Cartesian space.  $\dot{J} = \partial J / \partial t$  is differentiation of  $J$  with respect to time  $t$ . To simplify the description, Equation (3) is rewritten as follows:

$$J(\theta)\ddot{\theta} = \ddot{r}_a, \quad (4)$$

in which  $\ddot{r}_a = \ddot{r} - \dot{J}\dot{\theta}$ .

As the general form of robot dynamics, the single leg also is presented as follows:

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta}) + G(\theta) = \tau, \quad (5)$$

in which  $M(\theta)$ ,  $C(\theta, \dot{\theta})$ , and  $G(\theta)$  denote inertia matrix, centripetal force and Coriolis force, and the gravitational term, respectively.  $\tau = [\tau_1 \ \tau_2 \ \tau_3]^T$  is joint torques of the single leg.

It has been analyzed that a single optimization criterion is not suitable for legged robots that alter between support phase and flight (or swing) phase when walking or running. Then a modified optimization criterion that minimizes two norms including joint velocity and torques was proposed in our previous paper as follows:

$$\min \begin{cases} \frac{1}{2} (\alpha_s \dot{\theta}^T V \dot{\theta} + \beta_s \tau^T W \tau), & \text{support phase} \\ \frac{1}{2} (\alpha_f \dot{\theta}^T V \dot{\theta} + \beta_f \tau^T W \tau), & \text{flight phase,} \end{cases} \quad (6)$$

with equality constraints Equations (4) and (5) and inequality constraints, i.e., physical constraints of the joint as follows:

$$\begin{aligned} \theta^- &\leq \theta \leq \theta^+ \\ \dot{\theta}^- &\leq \dot{\theta} \leq \dot{\theta}^+ \\ \ddot{\theta}^- &\leq \ddot{\theta} \leq \ddot{\theta}^+, \end{aligned} \quad (7)$$

$\alpha_s$  and  $\beta_s$  are weight factors in support phase, while  $\alpha_f$  and  $\beta_f$  are weight factors in flight (or swing) phase. The values of these weight factors fall in the interval  $[0 \ 1]$  and satisfy  $\alpha_s + \beta_s = 1$ ,  $\alpha_f + \beta_f = 1$ .  $V$  and  $W$  are diagonal weight matrices reflecting effects of different joints.

The optimization problem formed by Equations (4)-(7) is converted into a standard QP problem on the joint angular acceleration level using the method base on ZNN in [27] as follows:

$$\min \begin{cases} \frac{1}{2} \ddot{\theta}^T Q_s \ddot{\theta} + p_s^T \ddot{\theta}, & \text{support phase} \\ \frac{1}{2} \ddot{\theta}^T Q_f \ddot{\theta} + p_f^T \ddot{\theta}, & \text{flight phase,} \end{cases} \quad (8)$$

subjected to equality and inequality constraints

$$\ddot{r}_a = J\ddot{\theta}, \quad \ddot{\theta}^- \leq \ddot{\theta} \leq \ddot{\theta}^+, \quad (9)$$

in which

$$Q_s = \alpha_s V + (1 - \alpha_s) M^T W M \quad (10)$$

$$p_s^T = \alpha_s \lambda V \dot{\theta} + (1 - \alpha_s) M^T W^T (C + G),$$

$$Q_f = \alpha_f V + (1 - \alpha_f) M^T W M \quad (11)$$

$$p_f^T = \alpha_f \lambda V \dot{\theta} + (1 - \alpha_f) M^T W^T (C + G),$$

$\lambda$  is a positive constant corresponding to convergence rate of ZNN.

The standard QP problem formulated by Equations (8) and (9) is equivalent to a kind of neural network named LVI-PDNN (LVI-based primal-dual neural network) as follows:

$$\begin{aligned} \dot{y} &= \begin{cases} \eta(I + H_s^T) \{P_\Omega [y - (H_s y + q_s)] - y\}, & \text{support phase} \\ \eta(I + H_f^T) \{P_\Omega [y - (H_f y + q_f)] - y\}, & \text{flight phase,} \end{cases} \end{aligned} \quad (12)$$

where  $y = [\ddot{\theta}^T, u^T]^T$ ,  $q_s = [p_s, -\ddot{r}_a^T]^T$ ,  $q_f = [p_f, -\ddot{r}_a^T]^T$ ,  $H_s = \begin{bmatrix} Q_s & -J_s^T \\ J_s & 0 \end{bmatrix}$ ,  $H_f = \begin{bmatrix} Q_f & -J_f^T \\ J_f & 0 \end{bmatrix}$ , and  $u$  denotes the dual decision variable vector consisting of coefficients of equality constraints in Equation (9).  $\eta$  is a positive constant that determines convergence rate of LVI-PDNN and should be selected large enough.  $P_\Omega(\cdot)$  is a piecewise linear projection operator as follows:

$$\begin{aligned} P_\Omega(\cdot) : R^{n+m+l} &\longrightarrow \Omega = \{y \mid \zeta^- \leq y \leq \zeta^+\} \\ [P_\Omega(y)]_i &= \begin{cases} \zeta_i^-, & y_i < \zeta_i^- \\ y_i, & \zeta_i^- \leq y_i \leq \zeta_i^+ \\ \zeta_i^+, & y_i > \zeta_i^+, \end{cases} \end{aligned} \quad (13)$$

$$i = 1, \dots, n + m + l.$$

### 3. CLVI-PDNN and Mid-Value CLVI-PDNN

In our previous paper [31] two kinds of neural network are proposed based on LVI-PDNN, i.e., CLVI-PDNN and mid-value CLVI-PDNN. CLVI-PDNN improves the drawback of

open-loop LVI-PDNN such as divergency of the optimized results from the desired value. Mid-value CLVI-PDNN further solves the problem that the optimized joint angles violate the physical constraints of the mechanism and deviate from the desired configuration especially for a legged robot.

CLVI-PDNN means closed-loop LVI-PDNN that introduces the output to the input of LVI-PDNN as feedback shown in Figure 3. In Figure 3, the block diagram in the blue dotted line shows LVI-PDNN proposed by Zhang [27]. The block diagram outside of the blue dotted line is a closed loop introducing feedback to the input of the neural network system as follows:

$$\begin{aligned} \ddot{r}_a &= \ddot{r}_d + k_{pr} (r_d - f(\theta)) + k_{ir} \int_0^t (r_d - f(\theta)) dt \\ &+ k_{pv} (\dot{r}_d - J(\theta)\dot{\theta}) + k_{iv} \int_0^t (\dot{r}_d - J(\theta)\dot{\theta}) dt \\ &- \dot{J}(\theta)\dot{\theta}, \end{aligned} \quad (14)$$

in which  $r_d$ ,  $\dot{r}_d$ , and  $\ddot{r}_d$  denote the desired motion of the foot in Cartesian space, respectively, while  $k_{pr}$ ,  $k_{ir}$ ,  $k_{pv}$ , and  $k_{iv}$  are coefficient matrices.

CLVI-PDNN has a critical defect that the optimized joint angles exceed the joint physical limitations leading to its uselessness for motion control of the quadruped robot. To improve this weakness of CLVI-PDNN, it could be modified as follows: defining a slack variable  $\varepsilon = \theta - \theta_d$ , where  $\theta_d$  denotes the desired joint configuration of the single leg when walking or running.  $\varepsilon$  could keep stable using a simple feedback control law

$$\theta = \theta_0 + k_p (\theta_d - \theta) + \int_0^t \dot{\theta} dt, \quad (15)$$

in which  $\theta_0$  means initial values and  $k_p$  means positive coefficient matrix. To prevent joint angles from exceeding the joint physical constraints, value of  $\theta_d$  should be selected close to the mid-value of the joint motion range. That is also why this algorithm is named mid-value CLVI-PDNN. Figure 4 shows the block diagram of mid-value CLVI-PDNN.

As analyzed in our previous paper [31], the mid-value CLVI-PDNN just compromise between the inequality and equality constrains. That is to say, the output of the QP solver maintains in motion range of the joints ensuring a feasible rather than optimal solution on condition that the error between the desired and the optimized trajectory is not enough to destroy normal motion control. To some extent, in many situations especially when the quadruped robots walk or run in the field, the output of mid-value CLVI-PDNN would be useful for motion control.

For some circumstances, such as high dimensions of the augmented matrix  $M_s$  and  $M_f$ , restricted performance of the onboard controller or computer, the discrete form of the QP solvers mid-value CLVI-PDNN may be more applicable for real-time control. We will deduce the discrete form of mid-value CLVI-PDNN and analyze the characteristics in next section.

#### 4. Discretize Mid-Value CLVI-PDNN

To solve the problem that continuous mid-value CLVI-PDNN is difficult to be applied in real-time control system, two kinds of discretization methods are presented in this section. For a general unconstraint optimization problem

$$\min_{x(t) \in R^n} f(x(t), t). \quad (16)$$

Suppose the extreme value of equation as follows:

$$\begin{aligned} \Omega_*(t) &= \left\{ (x^*(t), t) \mid g(x^*(t), t) = \frac{\partial f(x^*(t), t)}{\partial x^*(t)} = 0 \right\}. \end{aligned} \quad (17)$$

Obviously, condition of the extreme value is  $g(x(t), t) = 0$ , so that the error equation is selected as  $e(t) = g(x(t), t)$ . Then the error dynamics is designed on the basis of ZNN as follows:

$$\dot{e}(t) = -\lambda e(t), \quad (18)$$

where  $\lambda$  is a positive constant to scale the rate of convergency of  $e(t)$ . Substituting  $e(t) = g(x(t), t)$  into Equation (18) could obtain

$$H(x(t), t) \dot{x}(t) = -\lambda g(x(t), t) - g'(x(t), t), \quad (19)$$

where

$$\begin{aligned} H(x(t), t) &= \frac{\partial g(x(t), t)}{\partial x^T} = \frac{\partial^2 f(x(t), t)}{\partial x(t) \partial x^T(t)}, \\ g'(x(t), t) &= \frac{\partial g(x(t), t)}{\partial t} = \frac{\partial^2 f(x(t), t)}{\partial x(t) \partial t}. \end{aligned} \quad (20)$$

If the Hessian matrix  $H(x(t), t)$  is nonsingular, multiplying  $H^{-1}(x(t), t)$  both sides of Equation (19) results in

$$\dot{x}(t) = -H^{-1}(x(t), t) (\lambda g(x(t), t) + g'(x(t), t)). \quad (21)$$

Given initial value  $x(0)$ , the optimal state  $x^*(t)$  will be achieved on condition that Hessian matrix  $H(x(t), t)$  is positive definite. Actually, for our quadruped robot, Hessian matrix always maintains positive definite.

Taking forward Euler method as example, the discretization process could be presented as follows: rewriting Equation (21) as  $\dot{x} = G(x(t), t)$ , the forward Euler equation is as follows:

$$x_{k+1} = x_k + \delta G(x_k, t_k), \quad (22)$$

where  $x_k = x(t_k)$  means the value of  $x(t)$  at moment  $t_k$ .  $t_{k+1} = t_k + \delta$ , in which  $\delta$  denotes the sample interval. Therefore, discretization equation of  $x(t)$  is as follows:

$$x_{k+1} = x_k - H^{-1}(x_k, t_k) (\delta \lambda g(x_k, t_k) + \delta g'(x_k, t_k)). \quad (23)$$

Next the above-mentioned discretization process is applied for redundancy resolution of single leg. For a time-variant QP problem, it could be described at moment  $t_k$  as follows:

$$\begin{aligned} \min \quad & \frac{1}{2} \ddot{\theta}_k^T Q_k \ddot{\theta}_k + p_k^T \ddot{\theta}_k \\ \text{s.t.} \quad & J_L^k \ddot{\theta}_k = \ddot{r}_a^k \\ & \xi^- \leq \ddot{\theta}_k \leq \xi^+. \end{aligned} \quad (24)$$

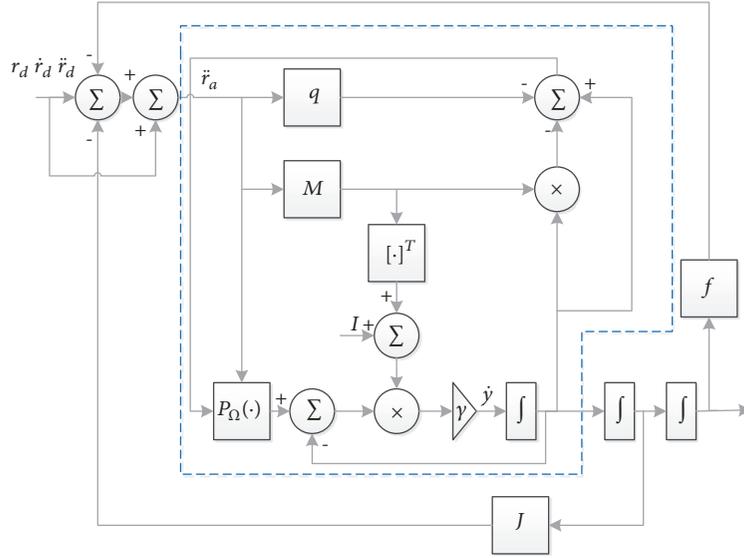


FIGURE 3: Block diagram of CLVI-PDNN [31].

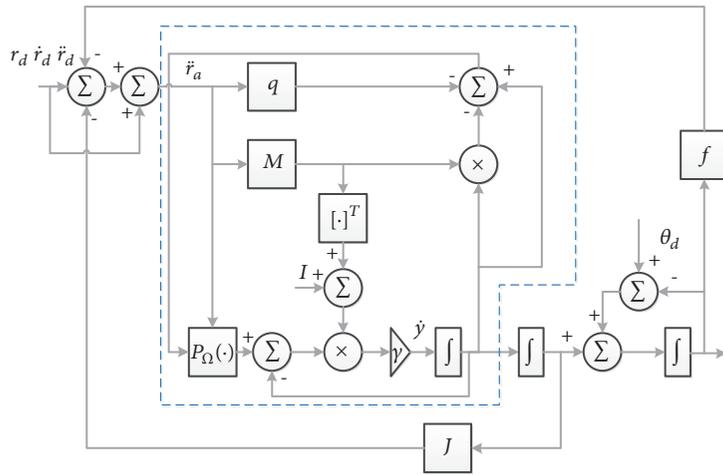


FIGURE 4: The block diagram of mid-value CLVI-PDNN [31].

Moreover, the dynamics of LVI-PDNN at moment  $t_k$  is as follows:

$$\dot{y}_k = \eta (I + H_k^T) \{P_\Omega [y_k - (H_k y_k + q_k)] - y_k\}, \quad (25)$$

where  $y_k = \begin{bmatrix} \ddot{\theta}_k \\ u_k \end{bmatrix}$ ,  $q_k = \begin{bmatrix} p_k \\ -\dot{r}_d^k \end{bmatrix}$ ,  $H_k = \begin{bmatrix} Q_k & -J_L^{kT} \\ J_L^k & 0 \end{bmatrix}$ .

For different discretization methods, accuracy and rapidity differ from each other. Taylor-type and bilinear transform-type discretization methods are used for mid-value CLVI-PDNN in this paper. The discretized state of neural network using bilinear transform is as follows:

$$y_{k+1} = y_k + \frac{\delta}{2} (\dot{y}_{k+1} + \dot{y}_k). \quad (26)$$

Substituting Equation (25) into Equation (26) obtains

$$\begin{aligned} y_{k+1} = & y_k + \frac{\delta}{2} \eta (I + H^T) \\ & \cdot (\{P_\Omega [y_{k+1} - (H_{k+1} y_{k+1} + q_{k+1})] - y_{k+1}\} \\ & + \{P_\Omega [y_k - (H_k y_k + q_k)] - y_k\}). \end{aligned} \quad (27)$$

Then, the recursion formula of  $y_k$  could be achieved:

$$\begin{aligned} y_{k+1} = & \frac{2 - \delta}{2 + \delta} y_k + \frac{\delta}{2 + \delta} \eta (I + H^T) \\ & \cdot (P_\Omega [y_{k+1} - (H_{k+1} y_{k+1} + q_{k+1})] \\ & + P_\Omega [y_k - (H_k y_k + q_k)]). \end{aligned} \quad (28)$$

$y_{k+1}$  in the right-hand side of Equation (28) could be estimated according to forward difference

$$\begin{aligned} y_{k+1} &= y_k + \delta \dot{y}_k \\ &= y_k \\ &\quad + \delta \eta \left( I + H^T \right) \{ P_\Omega [y_k - (H_k y_k + q_k)] - y_k \}, \end{aligned} \quad (29)$$

$$H_{k+1} = H_k + \delta \dot{H}_k = H_k + \Delta H_k,$$

$$q_{k+1} = q_k + \delta \dot{q}_k = q_k + \Delta q_k,$$

where  $\Delta H_k = H_k - H_{k-1}$  and  $\Delta q = q_k - q_{k-1}$ .

Taylor-type discretization method exploits high order terms of the Taylor series and then utilizes multiple points to discretize mid-value CLVI-PDNN. This kind of discretization method possesses the advantage of higher computational accuracy. Higher order corresponds higher accuracy while consuming more computation. Eliminating third-order derivative terms of Taylor series obtains

$$\dot{y}_k = \frac{2y_{k+1} - 3y_k + 2y_{k-1} - y_{k-2}}{2\delta} + O(\delta^2). \quad (30)$$

Substituting Equation (25) into Equation (30) achieves the recursion formula as follows:

$$\begin{aligned} y_{k+1} &= 1.5y_k - y_{k-1} + 0.5y_{k-2} + \delta \dot{y}_k \\ &= 1.5y_k - y_{k-1} + 0.5y_{k-2} \\ &\quad + \delta \eta \left( I + H_k^T \right) \{ P_\Omega [y_k - (H_k y_k + q_k)] - y_k \}. \end{aligned} \quad (31)$$

## 5. Theory Analysis

In this section, convergency and steady-state residual error of the discretization methods used in this paper is analyzed.

### 5.1. Bilinear Transform-Type Discretization Method

**Theorem 1.** *Bilinear transform-type recursion formula as Equation (28) is 0-stable.*

*Proof.* The characteristic polynomial of bilinear transform-type recursion formula as Equation (28) is

$$P(z) = z - \frac{2 - \delta}{2 + \delta} = 0, \quad (32)$$

which has a unique root on/in the unit disk, i.e.,  $z = (2 - \delta)/(2 + \delta)$  with  $0 < \delta < 1$ . Therefore, bilinear transform-type recursion formula as Equation (28) is 0-stable. The proof is thus completed.  $\square$

**Theorem 2.** *Bilinear transform-type recursion formula as Equation (28) is consistent and convergent, which converges with the order of truncation error  $O(\delta^2)$ .*

*Proof.* Discretization formula of  $y_k$  based on forward difference formula is as follows:

$$\dot{y}_k = \frac{y_{k+1} - y_k}{\delta} + O(\delta). \quad (33)$$

While discretization formula of  $y_k$  based on backward difference formula is as follows:

$$\dot{y}_{k+1} = \frac{y_{k+1} - y_k}{\delta} + O(\delta). \quad (34)$$

Adding Equations (33) and (34) could obtain the bilinear transform-type discretization formula

$$y_{k+1} = y_k + \frac{\delta}{2} (\dot{y}_{k+1} + \dot{y}_k) + O(\delta^2). \quad (35)$$

Note that dropping  $O(\delta^2)$  from Equation (35) yields exactly bilinear transform-type recursion formula as Equation (26), and, thus, the truncation error of Equation (26) is  $O(\delta^2)$ . Consequently, bilinear transform-type recursion formula as Equation (28) is consistent with order 2. Combining with *Theorem 1*, we know that the model in Equation (28) is both 0-stable and consistent, so that it can be conclusively concluded that bilinear transform-type recursion formula as Equation (28) is consistent and convergent, which converges with the order of its truncation error being  $O(\delta^2)$ . The proof is now completed.  $\square$

**Theorem 3.** *Considering the QP solver mid-value CLVI-PDNN based on ZNN, the steady-state residual error  $\lim_{k \rightarrow \infty} \|\dot{y}_k\|_2$  of bilinear transform-type discretization recursion formula as Equation (28) is  $O(\delta^2)$  in the sense of Euclidean norm.*

*Proof.* Let  $y_k^*$  denote the theoretical optimal solution of the QP problem described by Equations (8) and (9), we know that  $P_\Omega [y_k^* - (H_k y_k^* + q_k)] - y_k^* = 0$ . On the basis of *Theorems 1 and 2*, it could be concluded that  $y_k = y_k^* + O(\delta^2)$  with  $k$  large enough. Therefore, we obtain the limit error (the definition of error follows [42]):

$$\begin{aligned} \lim_{k \rightarrow \infty} \|\dot{y}_k\|_2 &= \lim_{k \rightarrow \infty} \left\| \eta \left( I + H_k^T \right) \right. \\ &\quad \cdot \left. \{ P_\Omega [y_k - (H_k y_k + q_k)] - y_k \} \right\|_2 = \lim_{k \rightarrow \infty} \left\| \eta \left( I \right. \right. \\ &\quad \left. \left. + H_k^T \right) \right. \\ &\quad \cdot \left. \{ P_\Omega [y_k^* + O(\delta^2) - (H_k (y_k^* + O(\delta^2)) + q_k)] \right. \\ &\quad \left. - y_k^* - O(\delta^2) \} \right\|_2 = \lim_{k \rightarrow \infty} \left\| \eta \left( I + H_k^T \right) \right. \\ &\quad \cdot \left. \{ P_\Omega [y_k^* - (H_k y_k^* + q_k)] - y_k^* \} \right. \\ &\quad \left. + \left( \eta \left( I + H_k^T \right) (I - H_k) - I \right) O(\delta^2) \right\|_2 \\ &= \lim_{k \rightarrow \infty} \left\| \left( \eta \left( I + H_k^T \right) (I - H_k) - I \right) O(\delta^2) \right\|_2. \end{aligned} \quad (36)$$

Considering that  $((I + H_k^T)(I - H_k) - I)$  is a constant matrix at a certain time instant, we further obtain

$$\begin{aligned} & \lim_{k \rightarrow \infty} \|\dot{y}_k\|_2 \\ &= \lim_{k \rightarrow \infty} \left\| \left( \eta (I + H_k^T) (I - H_k) - I \right) O(\delta^2) \right\|_2 \quad (37) \\ &= O(\delta^2). \end{aligned}$$

The proof is completed.  $\square$

**5.2. Taylor-Type Discretization Method.** Taylor-type recursion formula as Equation (30) is 0-stable. Taylor-type recursion formula as Equation (30) is consistent and convergent, which converges with the order of truncation error  $O(\delta^3)$ . These conclusions are proved in [42]. Therefore, the steady-state residual error could be concluded as follows.

**Theorem 4.** *Considering the QP solver mid-value CLVI-PDNN based on ZNN, the steady-state residual error  $\lim_{k \rightarrow \infty} \|\dot{y}_k\|_2$  of Taylor-type discretization recursion formula as Equation (31) is  $O(\delta^3)$  in the sense of Euclidean norm.*

*Proof.* Let  $y_k^*$  denote the theoretical optimal solution of the QP problem described by Equations (8) and (9), so that  $P_\Omega[y_k^* - (H_k y_k^* + q_k)] - y_k^* = 0$  and  $y_k = y_k^* + O(\delta^3)$  with  $k$  large enough. Therefore, we obtain the limit error as follows:

$$\begin{aligned} & \lim_{k \rightarrow \infty} \|\dot{y}_k\|_2 = \lim_{k \rightarrow \infty} \left\| \eta (I + H_k^T) \right. \\ & \cdot \left. \{P_\Omega[y_k - (H_k y_k + q_k)] - y_k\} \right\|_2 = \lim_{k \rightarrow \infty} \left\| \eta (I \right. \\ & + H_k^T) \\ & \cdot \left. \{P_\Omega[y_k^* + O(\delta^3) - (H_k(y_k^* + O(\delta^3)) + q_k)] \right. \\ & - y_k^* - O(\delta^3)\} \right\|_2 = \lim_{k \rightarrow \infty} \left\| \eta (I + H_k^T) \right. \quad (38) \\ & \cdot \left. \{P_\Omega[y_k^* - (H_k y_k^* + q_k)] - y_k^*\} \right. \\ & + \left. (\eta (I + H_k^T) (I - H_k) - I) O(\delta^3) \right\|_2 \\ &= \lim_{k \rightarrow \infty} \left\| (\eta (I + H_k^T) (I - H_k) - I) O(\delta^3) \right\|_2 \\ &= O(\delta^3). \end{aligned}$$

The proof is completed.  $\square$

## 6. Results and Discussion

The proposed discrete QP algorithms are operated in Matlab to solve the optimization problem formulated by Equations (8) and (9). In the simulation environment, the quadruped robot moves along horizontal direction with trotting gait in sagittal plane. The body of the quadruped robot moves at a constant speed on horizontal direction as well as constant height. The parameters of trotting gait are listed in Table 1.

TABLE 1: Parameters of trotting gait.

parameter	value	unit
$h$	0.7	m
$h_{apex}$	0.15	m
$v$	6.0	Km/h
$T$	0.6	s

TABLE 2: Physical parameters of single leg.

parameter	value	unit
$M$	50	kg
$M_1$	3	kg
$M_2$	3	kg
$M_3$	3	kg
$L_1$	0.25	m
$L_2$	0.30	m
$L_3$	0.30	m

TABLE 3: Physical constraints of joints.

parameter	value	unit
$\theta^+$	$(17\pi/36, 5\pi/6, 5\pi/6)$	rad
$\theta^-$	$(\pi/9, \pi/9, \pi/9)$	rad
$\dot{\theta}^+$	$(12, 12, 20)$	rad/s
$\dot{\theta}^-$	$-(12, 12, 20)$	rad/s
$\ddot{\theta}^+$	$(1700, 2200, 2800)$	rad/s
$\ddot{\theta}^-$	$-(1700, 2200, 2800)$	rad/s

TABLE 4: Parameters of mid-value CLVI-PDNN.

parameter	value
$\lambda$	3600
$\eta$	$10^7$
$k_{pr}$	1000
$k_{ir}$	100000
$k_{pv}$	100000
$k_{iv}$	100000

$h$  is a constant meaning the height of the hip.  $h_{apex}$  is the maximum height of the foot when the quadruped robot is walking.  $v$  is a constant meaning the desired velocity of the body.  $T$  is the gait cycle period of trotting gait.  $g$  is the gravitational acceleration. Parameters of the single leg system with half mass of the whole body in the sagittal plane as shown in Figure 2 are listed in Table 2.  $M$  is mass of body of the single system, i.e., half mass of the whole body. The others mean the masses and lengths of the three links of the single leg in sagittal plane.

The joint physical constraints, i.e., the upper and lower bound of joint angles, angular velocity and acceleration are listed in Table 3. The parameters of the discretized mid-value CLVI-PDNN are listed in Table 4.

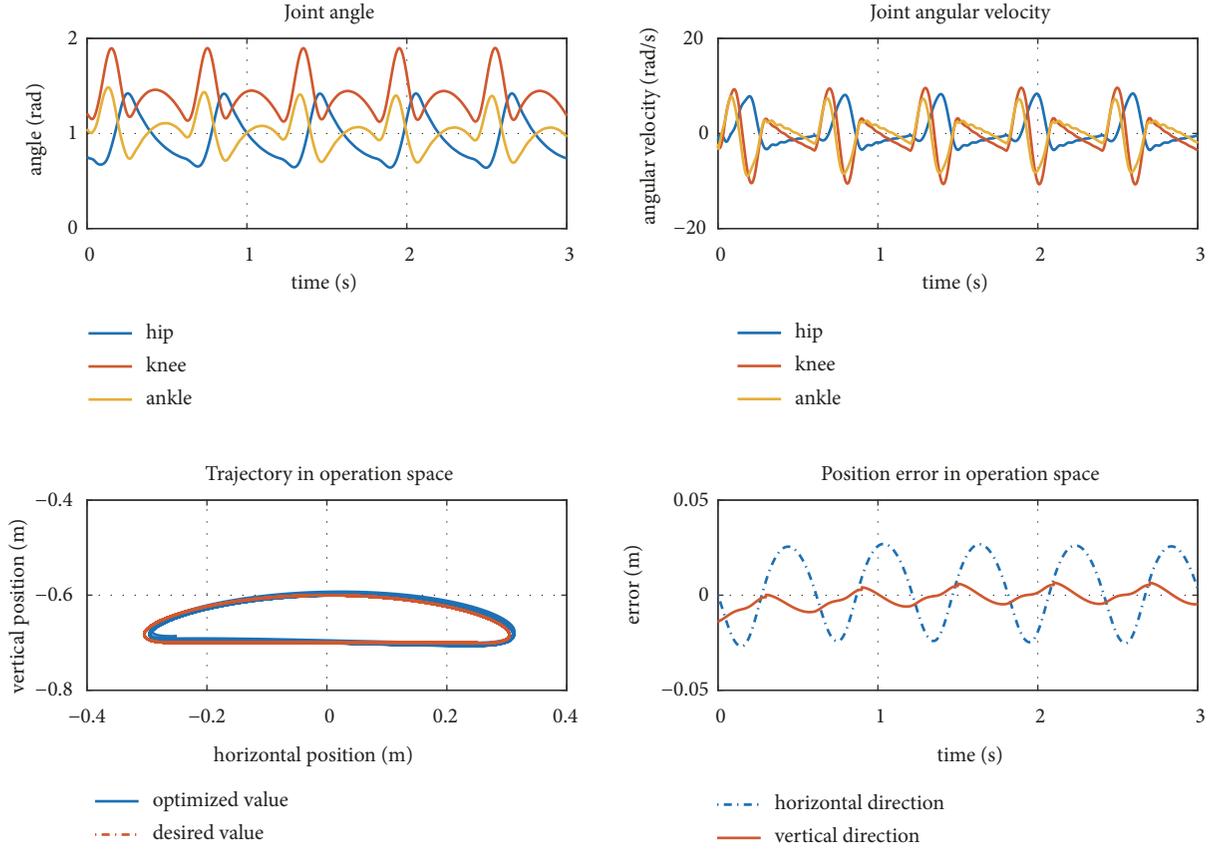


FIGURE 5: Taylor-type discrete mid-value CLVI-PDNN with parameter set 1.

In addition, motion trajectory of foot is designed as a curve respect to time  $t$  on the horizontal and vertical direction respectively. The expected motion curves in horizontal direction and vertical direction are described as follows:

$$x = \begin{cases} -\frac{\pi}{2} \sin\left(\frac{20\pi}{3}t\right) + \frac{5}{3}t - \frac{1}{4}, & 0 \leq t \leq 0.3 \\ \frac{1}{4} - \frac{5}{3}(t - 0.3), & 0.3 < t \leq 0.6, \end{cases} \quad (39)$$

$$y = \begin{cases} -0.7 + 237t^3 - 2370.4t^4 + 7901.2t^5 - 8779.1t^6, & 0 \leq t \leq 0.3 \\ -0.7, & 0.3 < t \leq 0.6. \end{cases} \quad (40)$$

The step length is set the same as the control cycle, i.e.,  $\delta = 1\text{ms}$ . The simulation results of the single leg under the aforementioned conditions are illustrated in Figures 5–10, in which Figures 5–7 present simulation results acquired by using Taylor-type method to discretize the mid-value CLVI-PDNN, while Figures 8–10 present simulation results acquired with bilinear transform-type discretization method.

The parameter set 1 in Figure 5 are  $V = W = \text{diag}([1, 1, 1])$  ( $\text{diag}(X)$  means a diagonal matrix whose diagonal elements are vector  $X$ ),  $\alpha_s = 0.9$ ,  $\alpha_f = 0.1$ ,  $\theta_d =$

$(0.6379, 1.8059, 1.0472)$ ; the parameter set 2 in Figure 6 are  $V = W = \text{diag}([1, 1, 1])$ ,  $\alpha_s = 0.1$ ,  $\alpha_f = 0.9$ ,  $\theta_d = (0.6379, 1.8059, 1.0472)$ ; the parameter set 3 in Figure 7 are  $V = \text{diag}([4, 2, 1])$ ,  $W = \text{diag}([1, 2, 4])$ ,  $\alpha_s = 0.1$ ,  $\alpha_f = 0.9$ ,  $\theta_d = (0.6379, 1.8059, 1.0472)$ . It is concluded that the optimized trajectory of the single leg in operation space could track the desired value well when using Taylor-type discretization method with position error about  $\pm 0.5\text{cm}$  in horizontal direction and  $\pm 2.5\text{cm}$  in vertical direction. For quadruped robot walking in the field that requires less

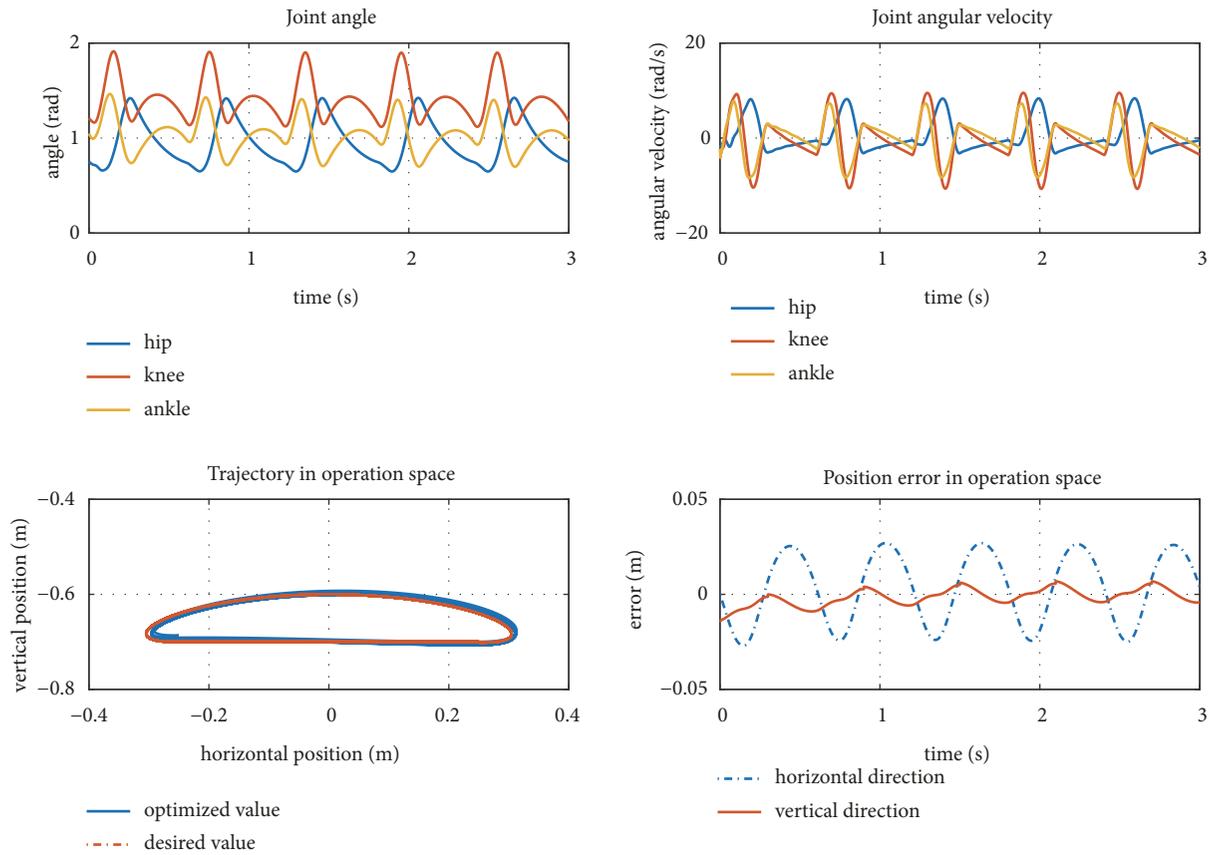


FIGURE 6: Taylor-type discrete mid-value CLVI-PDNN with parameter set 2.

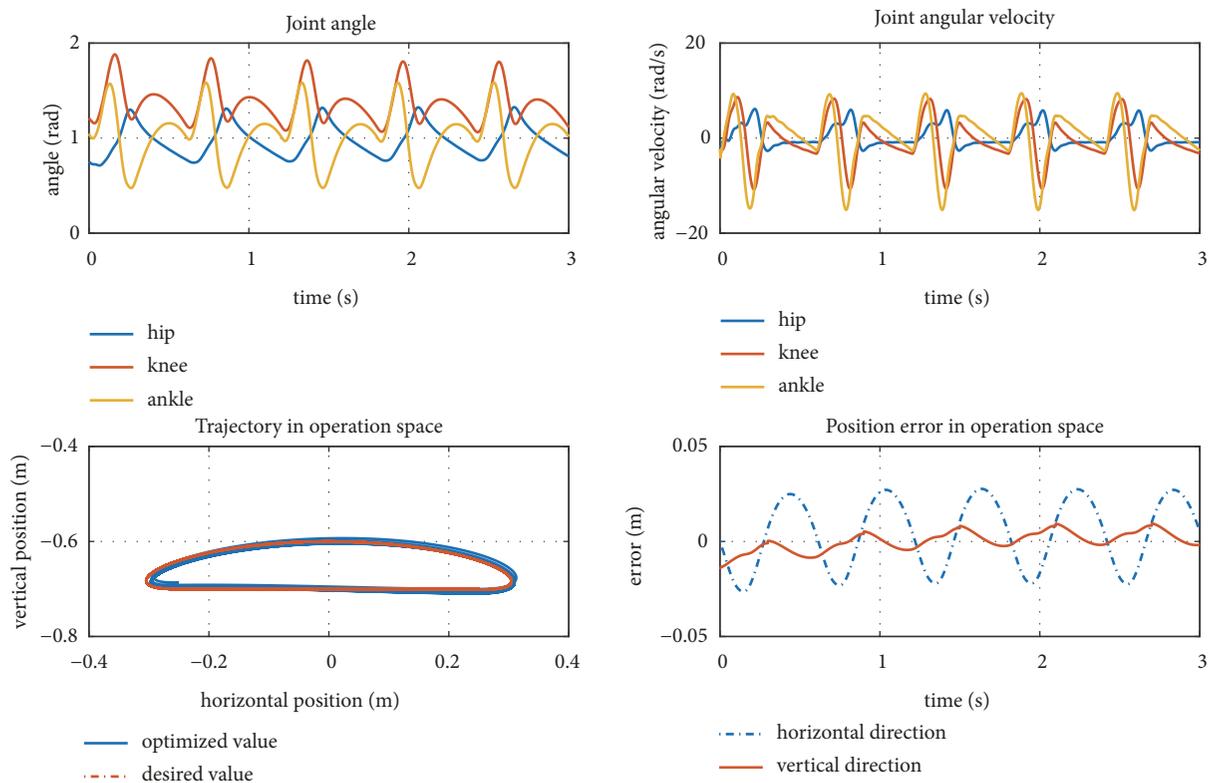


FIGURE 7: Taylor-type discrete mid-value CLVI-PDNN with parameter set 3.

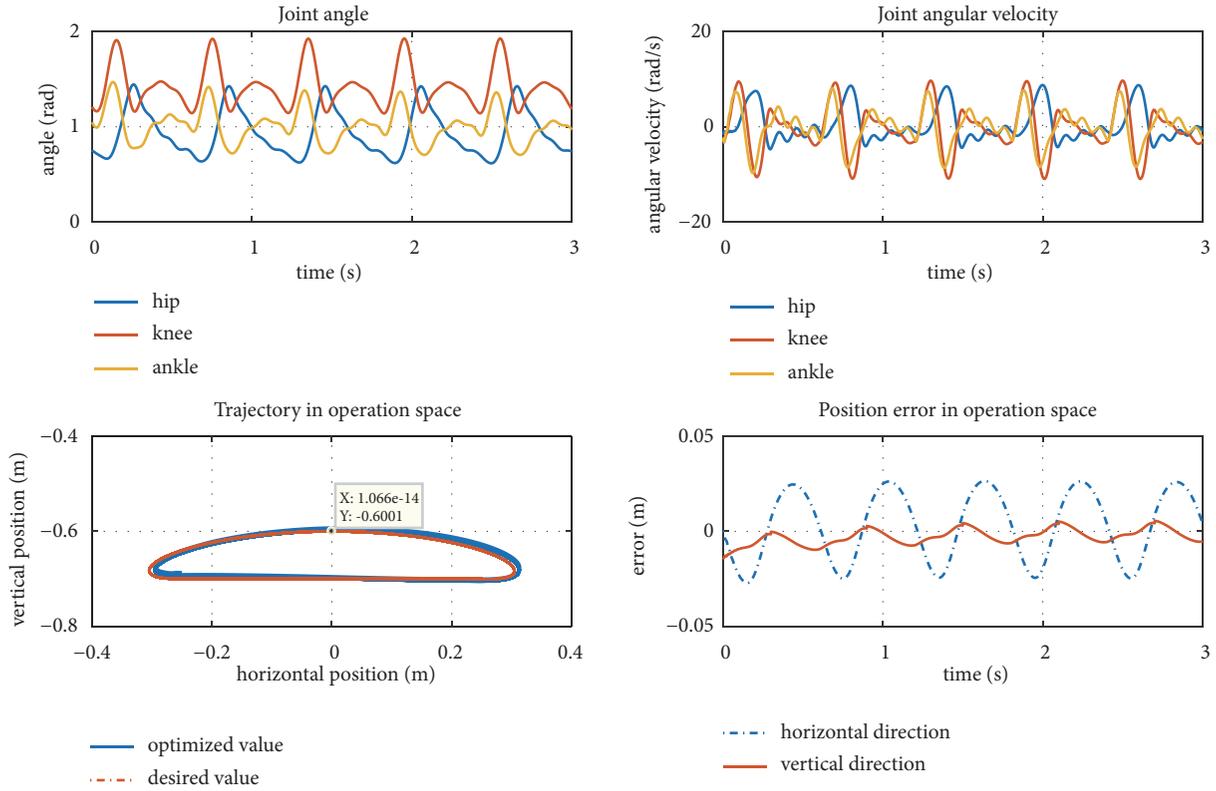


FIGURE 8: Bilinear transform-type discrete mid-value CLVI-PDNN with parameter set 1.

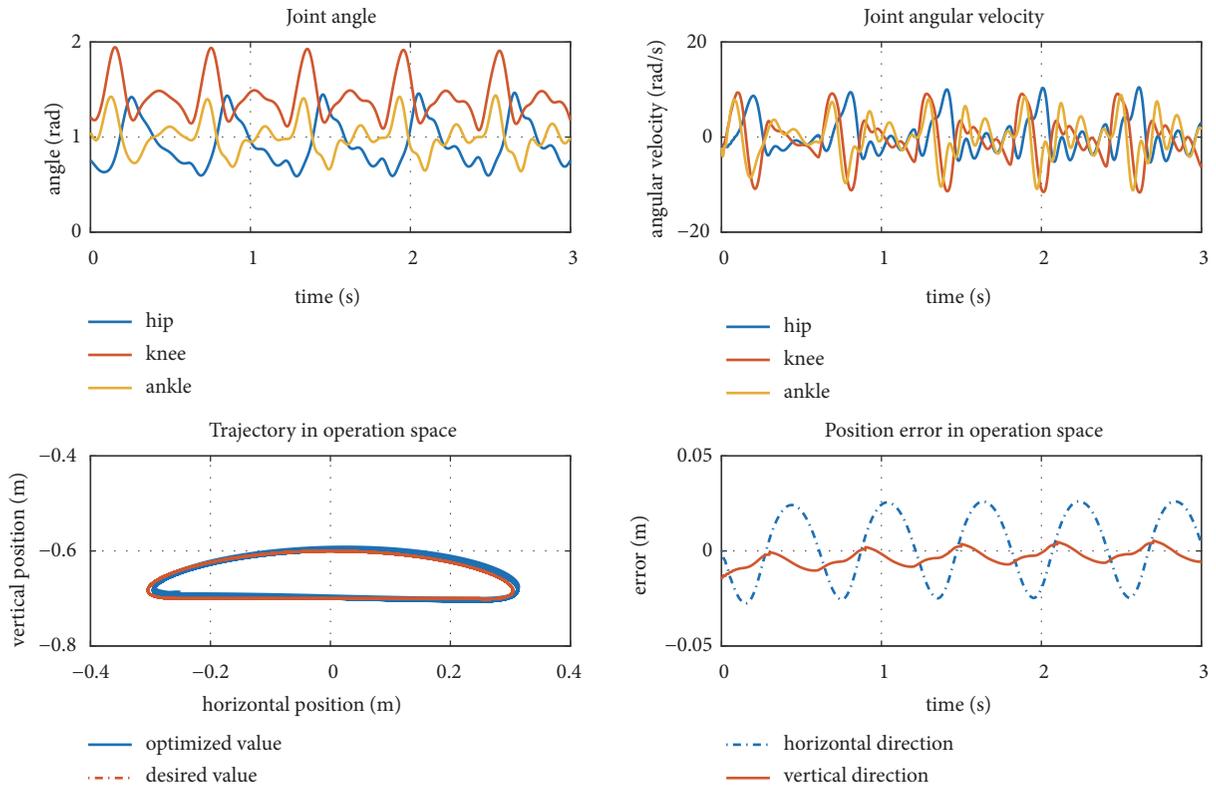


FIGURE 9: Bilinear transform-type discrete mid-value CLVI-PDNN with parameter set 2.

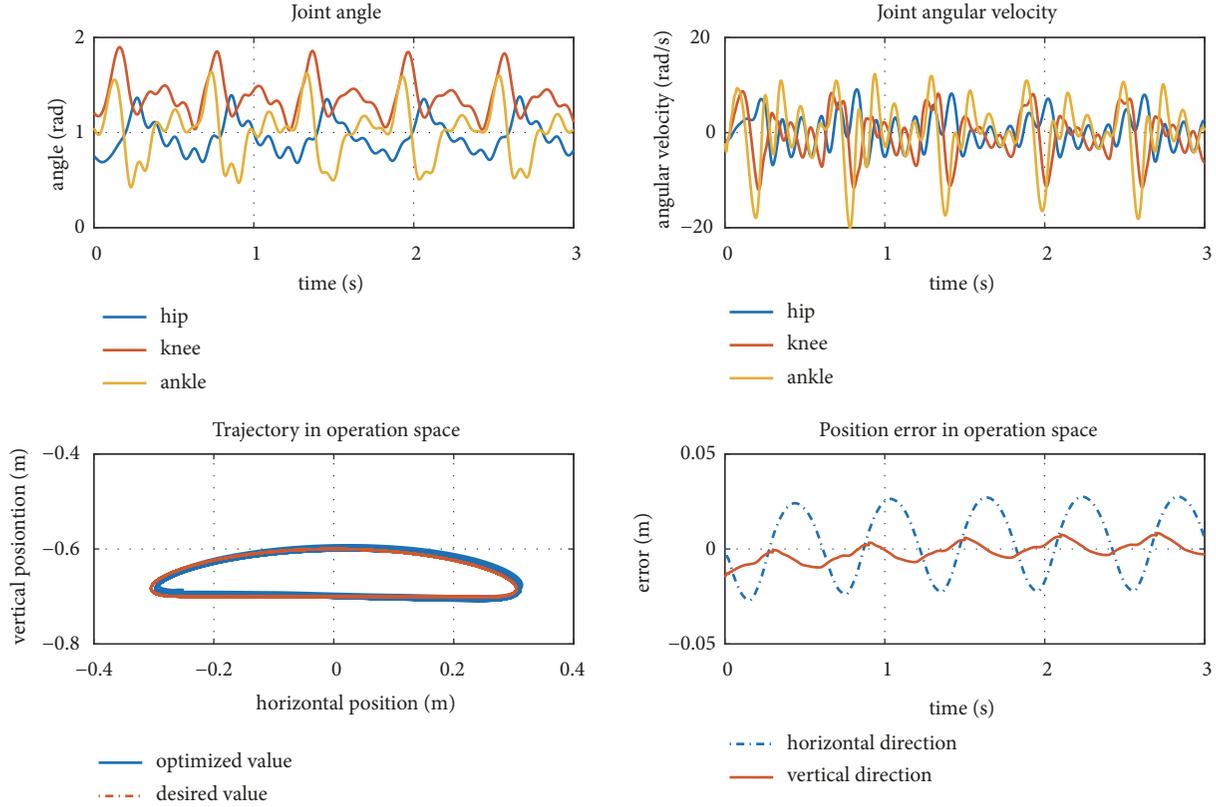


FIGURE 10: Bilinear transform-type discrete mid-value CLVI-PDNN with parameter set 3.

position precision, the position error shown in Figures 5–7 could satisfy the design and control requirements relative to the motion range of the single leg. On the other hand, different parameter sets result in different optimized motion curves in joint space. As illustrated in Figures 5 and 6, the joint angular velocities in Figure 6 with  $\alpha_s = 0.1$  are smoother than the curves in Figure 5 with  $\alpha_s = 0.9$  demonstrating that norm of minimizing joint torques would lead to instability. The angular velocity of hip joint with  $V = \text{diag}([1, 1, 1])$  shown in Figure 6 is larger than the value with  $V = \text{diag}([4, 2, 1])$  shown in Figure 7. While angular velocity of ankle joint shown in Figure 6 is smaller than the value shown in Figure 7. Moreover, the motion ranges of joint angles in Figures 6 and 7 are quite distinct correspondingly. The comparison between Figures 6 and 7 indicates the effect of different velocity weights among the three joints.

The parameter set 1 in Figure 8 are  $V = W = \text{diag}([1, 1, 1])$ ,  $\alpha_s = 0.9$ ,  $\alpha_f = 0.1$ ,  $\theta_d = (0.6379, 1.8059, 1.0472)$ ; the parameters set 2 in Figure 9 are  $V = W = \text{diag}([1, 1, 1])$ ,  $\alpha_s = 0.1$ ,  $\alpha_f = 0.9$ ,  $\theta_d = (0.6379, 1.8059, 1.0472)$ ; the parameters set 3 in Figure 10 are  $V = \text{diag}([4, 2, 1])$ ,  $W = \text{diag}([1, 2, 4])$ ,  $\alpha_s = 0.1$ ,  $\alpha_f = 0.9$ ,  $\theta_d = (0.6379, 1.8059, 1.0472)$ . It is concluded that the optimized trajectory of the single leg in operation space could track the desired value well using bilinear transform-type discretization method with position error about  $\pm 0.5\text{cm}$  in horizontal direction and  $\pm 2.5\text{cm}$  in vertical direction. But it is disappointing that the motion trajectory

in joint space is unsmooth and fluctuant so that perhaps it is difficult to be utilized in control of the single leg on the real system.

According to the simulation results of the two kinds of discretization methods, i.e., Taylor-type and bilinear transform-type, a conclusion could be achieved easily that the optimized motion trajectory in operation space tracks the desired value well and satisfies the control requirements. But the control of the single leg executes in joint space actually, the optimized motion trajectory in joint space by using Taylor-type discretization method is better than the other one obviously and easier to be applied in the control of the real mechanism.

## 7. Conclusion

As the discussion in our previous paper [31], we try to find a practicable QP solver to solve the inverse kinematics of the redundant single leg of the quadruped robot. In this paper we adopt two kinds of discretization methods to discretize the mid-value CLVI-PDNN proposed in our previous paper. According to the simulation results and analysis we could conclude that the optimized motion trajectories in operation space obtained by both Taylor-type and bilinear transform-type discretization methods track the desired values well if we do not overly concern about the position control precision of the single leg in the wild. Meanwhile, it is obvious that the optimized motion trajectory in joint space obtained

by Taylor-type discretization method is superior to the other one. Of course, higher order Taylor-type discretization methods will achieve more accurate inverse kinematics of the redundant single leg.

It was analyzed in our previous paper that the mid-value CLVI-PDNN QP solver just outputs feasible solutions rather than intuitional optimal solutions corresponding to the optimization criterion formulated in Equation (6). In addition, the deviation between the optimized value and the desired value is much larger than theory analysis. The error may be caused by the constraints such as integration threshold, the deviation between the criteria in Equations (6) and (8), and probably the crucial reason, i.e., the introduction of expected joint configuration. Although the optimized value deviates from the desired value to some extent, it is satisfactory to achieve a useable answer in the end.

### Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

### Conflicts of Interest

The authors declare no conflicts of interest.

### Acknowledgments

This research was funded by the Preschool Research Program Funds for the National University of Defense Technology (No. ZK17-03-49).

### References

- [1] S. Lee and A. K. Bejczy, "Redundant arm kinematic control based on parameterization," in *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, vol. 1, pp. 458–465, April 1991.
- [2] D. Tolani and N. I. Badler, "Real-time inverse kinematics of the human arm," *Presence: Teleoperators and Virtual Environments*, vol. 5, no. 4, pp. 393–401, 1996.
- [3] M. Shimizu, H. Kakuya, W.-K. Yoon, K. Kitagaki, and K. Kosuge, "Analytical inverse kinematic computation for 7-DOF redundant manipulators with joint limits and its application to redundancy resolution," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1131–1142, 2008.
- [4] G. K. Singh and J. Claassens, "An analytical solution for the inverse kinematics of a redundant 7dof manipulator with link offsets," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2976–2982, Taipei, Taiwan, 2010.
- [5] T. Zhang, Q. Wei, and H. Ma, "Position/force control for a single leg of a quadruped robot in an operation space," *International Journal of Advanced Robotic Systems*, vol. 10, no. 137, 2013.
- [6] H. Ding and S. P. Chan, "Real-time planning algorithm for obstacle avoidance of redundant robots," *Journal of Intelligent & Robotic Systems*, vol. 16, no. 3, pp. 229–243, 1996.
- [7] A. Khoukhi, L. Baron, and M. Balazinski, "A projected gradient augmented Lagrangian approach to multi-objective trajectory planning of redundant robots," *Transactions of the Canadian Society for Mechanical Engineering*, vol. 31, no. 4, pp. 391–405, 2007.
- [8] R. V. Mayorga and P. Sanongboon, "An artificial neural network approach for inverse kinematics computation and singularities prevention of redundant manipulators," *Journal of Intelligent & Robotic Systems*, vol. 44, no. 1, pp. 1–23, 2005.
- [9] K. A. O'Neil, "Divergence of linear acceleration-based redundancy resolution schemes," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 4, pp. 625–631, 2002.
- [10] J. A. Goble, Y. Zhang, Y. Shimansky, S. Sharma, and N. V. Dounskaia, "Directional biases reveal utilization of arm's biomechanical properties for optimization of motor behavior," *Journal of Neurophysiology*, vol. 98, no. 3, pp. 1240–1252, 2007.
- [11] K. Ohta, M. M. Svinin, Z. Luo, S. Hosoe, and R. Laboisière, "Optimal trajectory formation of constrained human arm reaching movements," *Biological Cybernetics*, vol. 91, no. 1, pp. 23–36, 2004.
- [12] Y. Zhang, J. Wang, and Y. Xia, "A dual neural network for redundancy resolution of kinematically redundant manipulators subject to joint limits and joint velocity limits," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 14, no. 3, pp. 658–667, 2003.
- [13] L. Righetti, J. Buchli, M. Mistry, M. Kalakrishnan, and S. Schaal, "Optimal distribution of contact forces with inverse-dynamics control," *International Journal of Robotics Research*, vol. 32, no. 3, pp. 280–298, 2013.
- [14] B. Siciliano and O. Khatib, *Springer Handbook of Robotics*, Springer, Berlin, Germany, 2008.
- [15] T. Szkodny and A. Legowski, "The decreasing of 3D position errors in the system of two cameras by means of the interpolation method," in *Proceedings of the International Conference on Intelligent Robotics and Applications*, vol. 9246, pp. 285–298, Springer, 2016.
- [16] Z. Nie, H. Xu, G. Yao, and L. Wu, "Study on interpolation algorithm of printing screen transferred manipulator," in *Advanced Graphic Communications, Packaging Technology and Materials*, vol. 369, pp. 675–681, Springer, New York, NY, USA, 2016.
- [17] M. Y. Kokurin, "Stable gradient projection method for non-linear conditionally well-posed inverse problems," *Journal of Inverse and ILL-Posed Problems*, vol. 24, no. 3, pp. 323–332, 2016.
- [18] J. Liu, C. Yang, J. Fang et al., "Improved optimization algorithm of the gram measurement matrix based on gradient projection," *Journal of Huazhong University of Science and Technology: Nature Science*, vol. 44, no. 8, pp. 62–65, 2016.
- [19] W. Jin, Y. Censor, and M. Jiang, "Bounded perturbation resilience of projected scaled gradient methods," *Computational Optimization And Applications*, vol. 63, no. 2, pp. 365–392, 2016.
- [20] B. Damas and J. Santos-Victor, "An online algorithm for simultaneously learning forward and inverse kinematics," in *Proceedings of the 25th IEEE/RSJ International Conference on Robotics and Intelligent Systems, IROS 2012*, pp. 1499–1506, Vilamoura, Portugal, October 2012.
- [21] O. Kanoun, F. Lamiroux, and P.-B. Wieber, "Kinematic control of redundant manipulators: Generalizing the task-priority framework to inequality task," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 785–792, 2011.
- [22] C. Chen and D. Jackson, "Parameterization and evaluation of robotic orientation workspace: A geometric treatment," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 656–663, 2011.

- [23] C. A. Pena, M. A. Guzman, and P. F. Cardenas, "Inverse kinematics of a 6 dof industrial robot manipulator based on bio-inspired multi-objective optimization techniques," in *Proceedings of the IEEE Colombian Conference on Robotics and Automation (CCRA)*, pp. 1–6, Bogota, Colombia, 2016.
- [24] Z.-G. Hou, L. Cheng, and M. Tan, "Multicriteria optimization for coordination of redundant robots using a dual neural network," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 40, no. 4, pp. 1075–1087, 2010.
- [25] Y. N. Zhang, "Analysis and design of recurrent neural networks and their applications to control and robotic systems," 2003.
- [26] S. Ma, "A balancing technique to stabilize local torque optimization solution of redundant manipulators," *Journal of Robotic Systems*, vol. 13, no. 3, pp. 177–185, 1996.
- [27] Y. Zhang, D. Guo, and S. Ma, "Different-level simultaneous minimization of joint-velocity and joint-torque for redundant robot manipulators," *Journal of Intelligent & Robotic Systems*, vol. 72, no. 3–4, pp. 301–323, 2013.
- [28] Y. Zhang, X. Lv, Z. Li, Z. Yang, and K. Chen, "Repetitive motion planning of PA10 robot arm subject to joint physical limits and using LVI-based primal-dual neural network," *Mechatronics*, vol. 18, no. 9, pp. 475–485, 2008.
- [29] Y. Zhang, W. Ma, X.-D. Li et al., "MATLAB simulink modeling and simulation of LVI-based primal-dual neural network for solving linear and quadratic programs," *Neurocomputing*, vol. 72, no. 7–9, pp. 1679–1687, 2009.
- [30] Y. Zhang and Z. Zhang, *Repetitive Motion Planning and Control of Redundant Robot Manipulators*, Springer, Heidelberg, Germany, 2013.
- [31] T. Zhang, H. An, and H. Ma, "Joint torque and velocity optimization for a redundant leg of quadruped robot," *International Journal of Advanced Robotic Systems*, vol. 14, no. 5, 2017.
- [32] M. J. Pérez-Illzarbe, "Convergence analysis of a discrete-time recurrent neural network to perform quadratic real optimization with bound constraints," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 9, no. 6, pp. 1344–1351, 1998.
- [33] M. J. Pérez-Illzarbe, "New discrete-time recurrent neural network proposal for quadratic optimization with general linear constraints," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 2, pp. 322–328, 2013.
- [34] H. Tang, H. Li, and Z. Yi, "A discrete-time neural network for optimization problems with hybrid constraints," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 21, no. 7, pp. 1184–1189, 2010.
- [35] Q. Liu and J. Cao, "Global exponential stability of discrete-time recurrent neural network for solving quadratic programming problems subject to linear constraints," *Neurocomputing*, vol. 74, no. 17, pp. 3494–3501, 2011.
- [36] Y. Zhang, W. Ma, and B. Cai, "From Zhang neural network to Newton iteration for matrix inversion," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 56, no. 7, pp. 1405–1415, 2009.
- [37] Y. Zhang, L. Xiao, G. Ruan, and Z. Li, "Continuous and discrete time Zhang dynamics for time-varying 4th root finding," *Numerical Algorithms*, vol. 57, no. 1, pp. 35–51, 2011.
- [38] D. Guo and Y. Zhang, "Zhang neural network, Getz-Marsden dynamic system, and discrete-time algorithms for time-varying matrix inversion with application to robots' kinematic control," *Neurocomputing*, vol. 97, pp. 22–32, 2012.
- [39] Y. N. Zhang, Y. W. Yang, B. H. Cai, and D. S. Guo, "Zhang neural network and its application to Newton iteration for matrix square root estimation," *Neural Computing and Applications*, vol. 21, no. 3, pp. 453–460, 2012.
- [40] Y. Zhang, B. Mu, and H. Zheng, "Link between and comparison and combination of Zhang neural network and quasi-Newton BFGS method for time-varying quadratic minimization," *IEEE Transactions on Cybernetics*, vol. 43, no. 2, pp. 490–503, 2013.
- [41] L. Jin and Y. Zhang, "Discrete-time Zhang neural network of  $O(\tau^3)$  pattern for time-varying matrix pseudoinversion with application to manipulator motion generation," *Elsevier Science Publishers*, 2014.
- [42] B. Liao, Y. Zhang, and L. Jin, "Taylor  $O(h^3)$  discretization of ZNN models for dynamic equality-constrained quadratic programming with application to manipulators," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 2, pp. 225–237, 2016.
- [43] L. Jin and Y. Zhang, "Discrete-time Zhang neural network for online time-varying nonlinear optimization with application to manipulator motion generation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 7, pp. 1525–1531, 2015.

