sensors

MDPI

Article

Simulation of Disturbance Recovery Based on MPC and Whole-Body Dynamics Control of Biped Walking

Xuanyang Shi [1,2], Junyao Gao [1,2,*], Yizhou Lu [1,2], Dingkui Tian [1,2] and Yi Liu [3]

[1] School of Mechatronical Engineering, Intelligent Robotics Institute, Beijing Institute of Technology, Beijing 100081, China; shixuanyang@bit.edu.cn (X.S.); 3120180169@bit.edu.cn (Y.L.); tiandingkui@bit.edu.cn (D.T.)
[2] Beijing Advanced Innovation Center for Intelligent Robots and Systems, Beijing 100081, China
[3] Beijing Institute of Astronautical Systems Engineering, Beijing 100076, China; yiliu@bit.edu.cn
[*] Correspondence: gaojunyao@bit.edu.cn

Received: 26 March 2020; Accepted: 21 May 2020; Published: 24 May 2020

Abstract: Biped robots are similar to human beings and have broad application prospects in the fields of family service, disaster rescue and military affairs. However, simplified models and fixed center of mass (COM) used in previous research ignore the large-scale stability control ability implied by whole-body motion. The present paper proposed a two-level controller based on a simplified model and whole-body dynamics. In high level, a model predictive control (MPC) controller is implemented to improve zero moment point (ZMP) control performance. In low level, a quadratic programming optimization method is adopted to realize trajectory tracking and stabilization with friction and joint constraints. The simulation shows that a 12-degree-of-freedom force-controlled biped robot model, adopting the method proposed in this paper, can recover from a 40 Nm disturbance when walking at 1.44 km/h without adjusting the foot placement, and can walk on an unknown 4 cm high stairs and a rotating slope with a maximum inclination of 10°. The method is also adopted to realize fast walking up to 6 km/h.

Keywords: MPC; whole-body dynamics; quadratic optimization; disturbance recovery

1. Introduction

Humanoid robots imitate the form of human biped walking with the aim to move in unstructured complex environments and overcome external disturbances [1,2]. Compared with wheeled and tracked robots, humanoid robots have a stronger adaptability in complex environments and a wide range of applications in many fields. A good performance for a walking robot can be simply defined as moving from a starting to a goal point without falling down within disturbance, which is usually caused by external force disturbance, uneven ground and unmodeled dynamics in high dynamic motion. In this sense, the main objective of the control in a walking robot is to guarantee zero moment point (ZMP) and body posture to keep the dynamic balance [3]. However, generating fast and stable walking under disturbance for humanoid robots is a multidisciplinary and complex subject, due to the naturally unstable dynamics of these types of robots. To reduce the difficulty of robot stability control, the overall structure of the walking system in preview works is usually decoupled into four hierarchy levels, which are footstep planner, reference generators, disturbance recovery and low-level controller [4]. The preview methods usually use preview control or differential dynamic program (DDP) to plan the offline trajectory, fix the center of mass (COM) to the waist, and use linear quadratic regulator (LQR) [5] controllers based on the linear inverted pendulum model (LIPM) to keep the robot's stability by keeping the ZMP inside the support polygon. The method controls the ZMP by adding a small center of mass (COM) trajectory adjustment amount. When the disturbance is too large, the control is easy to diverge and it is difficult to realize large disturbance recovery. However, the COM is not fixed to a point on the body during the robot's motion, but rather changes dynamically. Otherwise, the authors of this paper believe that stable walking can be realized by controlling whole-body instantaneous dynamics to ensure that ZMP at each control period is in the support area instead of perfect tracking.

In recent years, many authors have studied the subject of humanoid balance control. The key issue in the matter is to look for a fast algorithm that allows a control even if the robot is subjected to impulsive disturbances caused by external force or unknown uneven ground. Furusho [6] stabilize whole-body balance or posture by using the ankle torque of the support leg by modeling the whole robot as a simple inverted pendulum. Nagasaka introduced a torso position compliance (TPC) control method [7]. This control law is effective for a walking robot with feet of high stiffness and was used to stabilize humanoid robots H5 and H7 [8]. Hirai et al. proposed the model ZMP control method [9]. The main idea is when the body of the real robot is inclined more forward than the model, the model body is more strongly accelerated than the planned trajectory. This changes the target inertial force and the target ZMP then goes more backward than the original ZMP, hence producing posture recovery in the real robot. Yu et al. proposed a body acceleration and foot placement adjustment method to reject disturbance [10]. Li et al. proposed a novel control framework to demonstrate a unique foot tilting maneuver based on ankle torque control for humanoid balance recovery [11]. These methods are only based on the simplified model or certain parts and they are all based on the current optimality; they have achieved good results but are difficult to adapt to large disturbances.

When a human is in motion, most joints of the body participate in the movement to maintain balance within disturbance [12]. Using the whole body to maintain balance is better than the simplified model. Yoshino [13] model the walking robot as a multi-input-multi-output (MIMO) system having joint torques as its input vector and output as the state of all links. By applying a LQR controller, the robot could successfully walk at 3 km/h on the floor with an unevenness of 6 mm. This method needs to linearize the robot system, which will lose more characteristics of the robot. Khatib [14] first used inverse dynamics to map the task space control to the joint space control, calculating the joint acceleration or joint moment; this allows us to exert control at the task space and map to the joint space, indirectly achieving nonlinear control. This method relies on convex optimization to solve tracking and constraint problems. Hutter et al. treated the floating inverse dynamics problem as a quadratic programming (QP) problem and provided a method of solving inverse dynamics problems with constraints [15]. Stephane Caron et al. extend walking stabilization based on LIPM tracking by quadratic programming-based wrench distribution, and they showed that the HRP-4 was able to climb an industrial staircase with 18.5 cm-high steps [16]. Feng et al. applied the optimization-based inverse kinematic and inverse dynamic control method in atlas at the defense advanced research projects agency (DARPA) challenge [17]. However, the instantaneous stability is achieved only by whole-body control, and the pre-planned trajectory is not adjusted in real-time in the case of large-scale disturbances.

In this paper, a two-level controller of the simplified model and the whole-body model to reject disturbance, instead of using the simplified model and fixed COM, is proposed; the whole control is divided into two levels, the high level is motion planning control based on the LIP model, the low level is task space controller, including trajectory tracking, environment constraints and whole-body ZMP control based on whole-body dynamics (as shown in Figure 1). Firstly, we use the preview control method to generate the walking pattern. The motion planning section also generates reference arm motion trajectories and reference contact forces. In addition, we design a real-time MPC controller to stabilize ZMP. Secondly, a feedback and feedforward controller are applied to calculate the task space acceleration. Thirdly, a whole-body dynamics controller is implemented considering the friction constraint and joint constraint. The QP optimization method is adopted to map the task space acceleration and expected ZMP to joint torque. Finally, a physical simulation environment is established to verify the proposed method and to test external force disturbance, walking on uneven ground and fast walking.

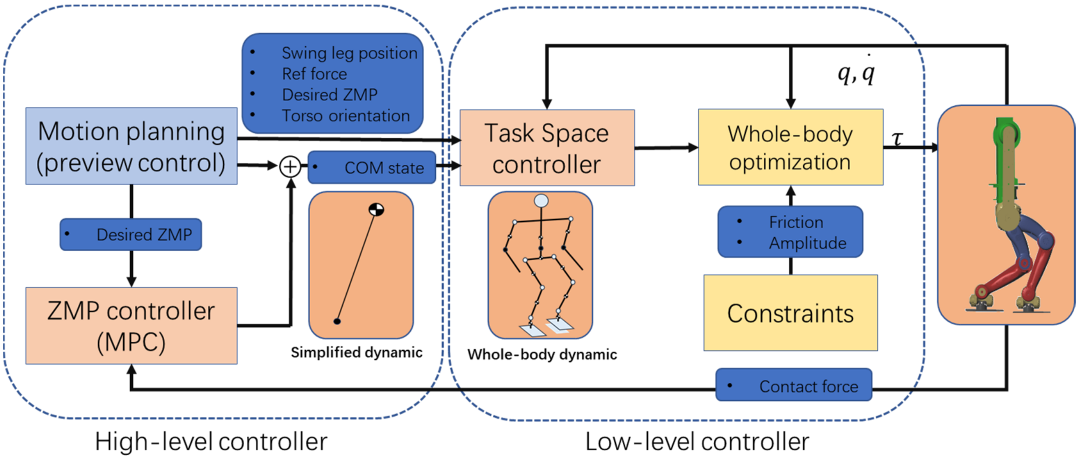

Figure 1. Block diagram of whole-body motion control. q and $\dot{q}$ are joints position and velocity, and τ is the joints torque.

There are three main contributions of this paper. The first is using MPC controller to stabilize ZMP to get future optimization, instead of LQR that considers current optimization. The second is using whole-body dynamics, which is more like the real robot dynamics, to get a wider range of disturbance rejection abilities. The third is that we considered the environment constraint to get better environmental adaptability by using QP optimization.

2. High-Level Trajectory Planning and Control

2.1. Motion Planning

This section mainly introduces the motion planning method, using the preview control method proposed by Kajita [5] to generate the walking pattern. The main principle is simplifying the robot as a linear inverted pendulum, fixing the vertical height of the COM and using the future foot placement information to plan the COM motion in advance, which is similar to model predictive control. Given the predefined foot placements with a gait cycle and step length, the desired ZMP is located in the support foot in the single support phase and moves to the next support foot in the double support phase by a cubic interpolation curve. Then, given the COM height, the contact force and COM trajectory, including position and velocity, can be calculated.

Figure 2 shows the linear inverted pendulum model of the humanoid robot. Equations (1) and (2) are the dynamical system of 3D LIPM. State x is the COM position, u is the control input, z_c is the COM height, p_x is ZMP and $X = \begin{bmatrix} x & \dot{x} & \ddot{x} \end{bmatrix}^T$ is the state vector.

$$\frac{d}{dt}\begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u \tag{1}$$

$$p_x = \begin{bmatrix} 1 & 0 & z_c/g \end{bmatrix}\begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} \tag{2}$$

The cost function J of the preview controller and control input $u(k)$ of walking is:

$$J = \sum_{i=k}^{\infty}\left\{Q_e e(i)^2 + \Delta X^T(i) Q_X \Delta X(i) + R_X \Delta u(i)^2\right\} \tag{3}$$

$$u(k) = -G_i \sum_{i=0}^{k} e(k) - G_x X(k) - \sum_{j=1}^{N_L} G_p(j) p^{ref}(k+j) \tag{4}$$

where $e(i) = p(i) - p^{ref}(i)$ is the ZMP error, $\Delta X(i)$ is the state change increment, $\Delta u(i)$ is the control input, G_i is the ZMP feedback gain, G_X is the state feedback gain and $G_p(j)$ is the future ZMP feedforward gain. Q_e, Q_X and R_X are the weight matrix. The control input contains not only the feedback of the current state and the current ZMP error but also reference ZMP information for N_L future control cycles. According to the reference ZMP information for the future, the controller can predict the future COM trajectory in advance, so that the COM motion velocity and acceleration will not be too large when the ZMP changes. The preview control weights are $Q_e = 1$, $R_x = 10e - 8$, $Q_x = 0$ and $N_L = 500$. The control gains can be calculated off-line by using MATLAB function dare to solve the Riccati equation. A sixth-order curve interpolation is used to generate the trajectory of the swinging leg.

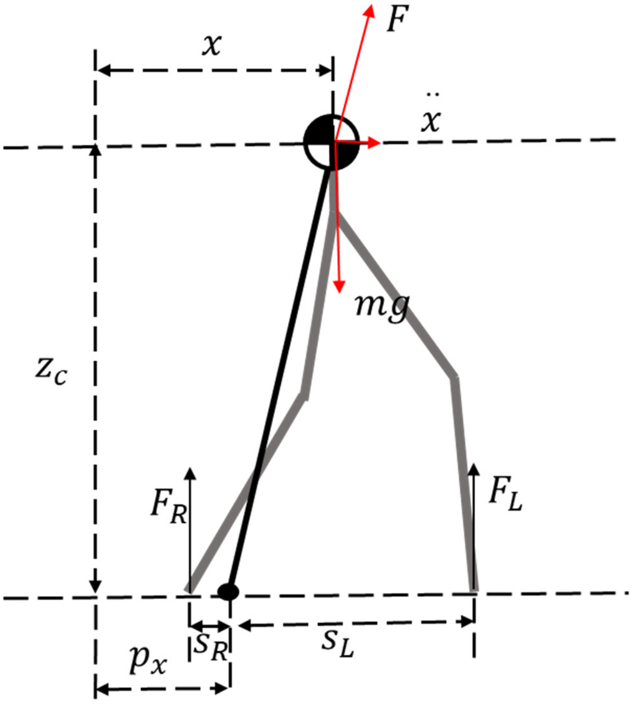

Figure 2. The linear inverted pendulum model of the humanoid robot.

In the single support phase, the contact force of the support foot in the z direction is mg, the swing foot is 0. In the double support phase, the contact force can be calculated by the foot position and ZMP. The contact forces satisfy the following relationships as:

$$\begin{cases} F_R = \frac{s_L}{s_R+s_L} mg \\ F_L = \frac{s_R}{s_R+s_L} mg \end{cases}, \tag{5}$$

Adopting this preview control method, the COM trajectory, foot trajectory and reference contact force are obtained.

2.2. Model Predictive Control of ZMP

Thinking about a stabilization method based on a LIP model. In this case we must measure the ZMP to design a feedback controller. As mentioned in [7,8], previous researchers used LQR to control the acceleration of the center of mass to stabilize ZMP. However, LQR only pursues the current optimal, resulting in a relatively large control input. Thus, an MPC controller is implemented in this paper to improve the control performance.

The ZMP equation calculated by the LIP model is:

$$^{B}p_{x,cal} = x - \frac{z_c}{g}\ddot{x}, \tag{6}$$

Define the measured ZMP as $^{B}p_{x,rel}$, and, by assuming the time constant of the ZMP sensor as T, the relationship between measured ZMP and calculated ZMP is:

$$^{B}p_{x,rel} = \frac{1}{1+Ts}\,^{B}p_{x,cal}, \tag{7}$$

Rewrite this as an error state function as (8), $\Delta X_c = \left[^{B}\Delta p_{x,rel}, \Delta x, \Delta\dot{x}\right]^T$ is the error state and $u_c = \Delta\ddot{x}$ is the control input. It is noteworthy that the control quantity is an additional acceleration on the original trajectory.

$$\begin{gathered}
\frac{d}{dt}\begin{bmatrix} ^{B}\Delta p_{x,rel} \\ \Delta x \\ \Delta\dot{x} \end{bmatrix} = \begin{bmatrix} -\frac{1}{T} & \frac{1}{T} & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} ^{B}\Delta p_{x,rel} \\ \Delta x \\ \Delta\dot{x} \end{bmatrix} + \begin{bmatrix} -\frac{z_c}{gT} \\ 0 \\ 1 \end{bmatrix}\Delta\ddot{x}\,, \\
^{B}\Delta p_{x,rel} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} ^{B}\Delta p_{x,rel} \\ \Delta x \\ \Delta\dot{x} \end{bmatrix},
\end{gathered} \tag{8}$$

To compute a feasible control input, we solve them through model predictive control as Equation (9). Here we sample all time-varying quantities at N knot points, with the time durations, h, A_c and B_c being the state matrix after discretization; Q_c and R are the weight coefficients. At time zero, given $\Delta x = 0$ and $\Delta\dot{x} = 0$, and it is calculated by integration of the control input. The $^{B}\Delta p_{x,rel}$ is updated in each control cycle based on the desired ZMP and the measured ZMP. Add the values Δx, $\Delta\dot{x}$ and $\Delta\ddot{x}$ onto the previously generated trajectory as inputs to the whole-body controller. The optimization problem can be solved in under 1 ms.

$$\begin{aligned}
\underset{u_c(k)}{min} \quad & \sum_{k=1}^{N}\left(\Delta X_c^T(k) Q_c \Delta X_c^T(k) + R\Delta u_c(k)^2\right) \\
st. \quad & \Delta X_c^T(k+1) = \quad A_c \cdot \Delta X_c^T(k) + B_c \cdot \Delta u_c(k)
\end{aligned} \tag{9}$$

The method is numerically simulated to verify the stability. The ParNMPC [18] solver is used to solve this optimization problem. The parameters are N = 12, h = 0.004 s and the control period is 1 ms. The time delay of the force sensor is 50 ms, measured in the previous experiment, and at t = 1 s, a disturbance is applied to the state variable $^{B}\Delta p_{x,rel}$ had an amplitude of 0.15 m, which is the maximum ZMP error that can be obtained according to the size of foot. Q_c is set to $diag([1,1,1])$ and R is set to 10^{-2}, 10^{-4}, 10^{-6} and 10^{-8} to test the performance of different weight coefficient. As shown in Figure 3a. When ZMP error occurs in the system, the controller can quickly eliminate the ZMP error within 0.1 s. Figure 3b,c show that the states Δx and $\Delta\dot{x}$ also go back to equilibrium point 0. By comparing different R parameters, when R is larger, the control quantity u is smaller, resulting in the slow decrease of ZMP error, as shown in Figure 3d. Thus, different control effects can be produced by adjusting R.

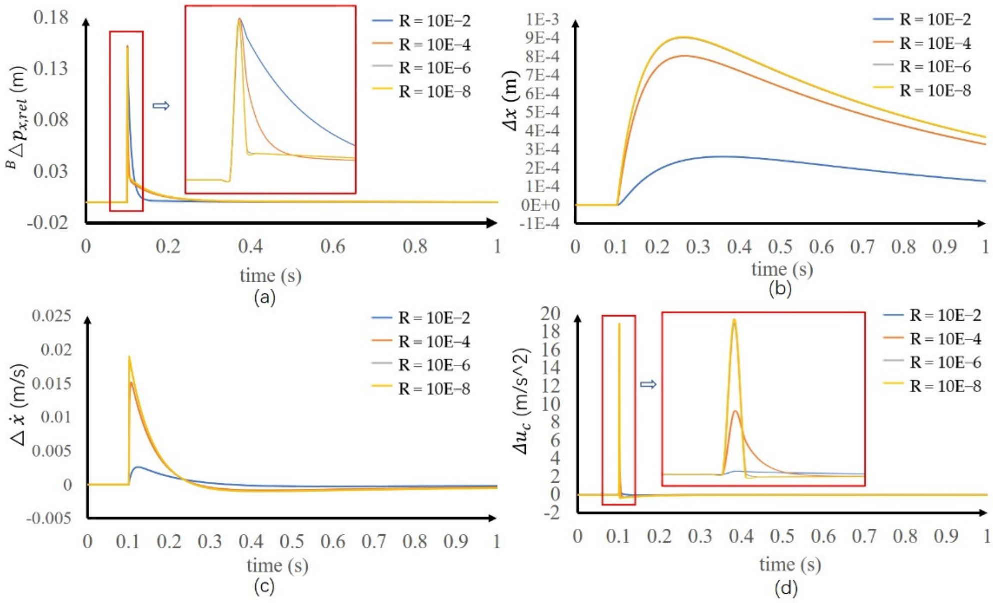

Figure 3. Simulation results of zero moment point (ZMP) error control. (**a**) ZMP error. (**b**) COM position increment. (**c**) COM velocity increment. (**d**) Control input.

3. Whole-Body Controller

3.1. Modeling

The degrees of freedom (DoFs) layout of the humanoid robot used in this article is shown in Figure 4. There are 18 DoFs in total, including two DoFs on the shoulders, 10 DoFs on the legs and six floating DoFs. The feet of the humanoid robot are not firmly connected to the ground and only provide a supporting force with limited torque. The floating coordinate system is established and fixed to the waist of the robot. $q_b = \left\{q_{bz}, q_{by}, q_{bx}, w_{bz}, w_{by}, w_{bx}\right\}$ is the generalized joint variable of the floating base, which represents the position and Euler angle of the floating base viewed in the world coordinate frame. $q_r = \{q_{r1}, q_{r2}, q_{r3}, q_{r4}, q_{r5}\}$ and $q_l = \{q_{l1}, q_{l2}, q_{l3}, q_{l4}, q_{l5}\}$ are, respectively, the generalized coordinates of the active joints of the right and left legs. $q_{arm} = \{q_{arm1}, q_{arm2}\}$ denotes the arm joint variables. The robot's active joints and floating joints are denoted as the robot's generalized joint coordinate variable $q = \{q_b, q_r, q_l, q_{arm}\}$, which is an 18 × 1 vector. Each link of the robot has mass and inertia.

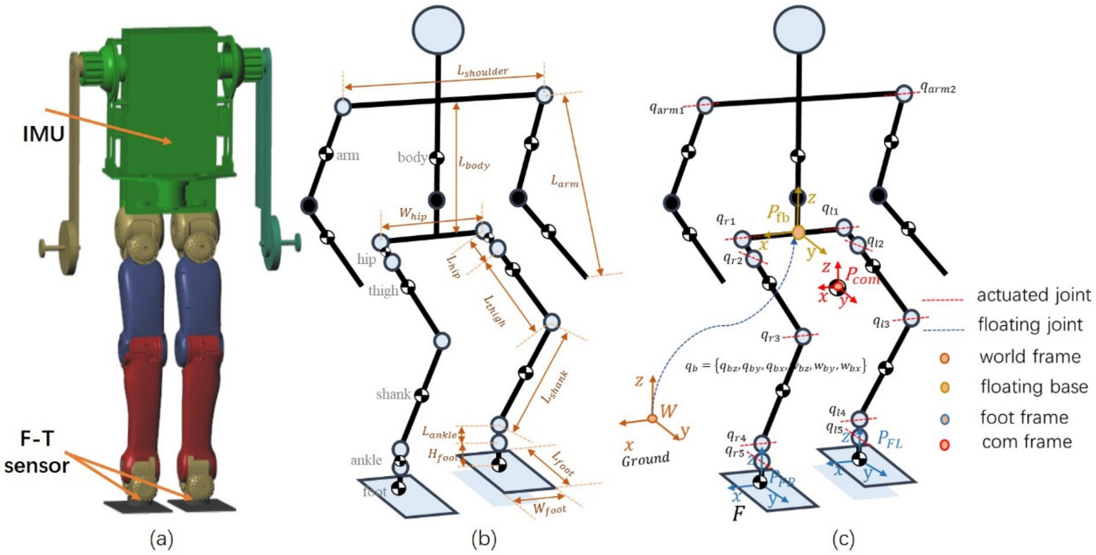

Figure 4. Typical motor-driven humanoid robot. They always have an inertial measurement unit (IMU) fixed inside the chest and a six-dimensional force torque (F-T) sensor for each ankle. Each leg has five degrees of freedom (DoFs): (**a**) Simulation model; (**b**) model mechanical parameters; (**c**) DoFs configuration.

A single-support phase, double-support phase and flight phase may appear during robot motion. For convenience of description, dynamics modeling is established for the double-support phase. The single-support phase and flight phase simply require the contact force to be set to zero. The standard form of robotic body dynamics is:

$$D(q)\ddot{q} + C\left(q, \dot{q}\right)\dot{q} + G(q) = B\tau + J^T(q)F, \tag{10}$$

To avoid the construction of a contact dynamics model and direct control of the contact force, the dynamic equation is rewritten as:

$$\begin{bmatrix} D(q) & -B & -J^T(q) \end{bmatrix} \cdot X + C\left(q, \dot{q}\right)\dot{q} + G(q) = 0, \tag{11}$$

In Equation (11), $X = \left[\ddot{q}, \tau, F\right]^T \in \mathfrak{R}^{48\times 1}$ is the generalized state variable of the robot while $\ddot{q} \in \mathfrak{R}^{18\times 1}$ and $\tau \in \mathfrak{R}^{18\times 1}$ are, respectively, the joint acceleration and joint torque. $F \in \mathfrak{R}^{12\times 1}$ is the contact-force variable. The contact-force variable includes two contact points of the left and right feet. Each contact point includes contact forces and contact moments in three directions. $D(q) \in \mathfrak{R}^{18\times 18}$ is the inertia matrix. $B \in \mathfrak{R}^{18\times 18}$ is the joint torque mapping matrix. $J^T(q) \in \mathfrak{R}^{18\times 12}$ is the Jacobian matrix of the contact point. $C\left(q, \dot{q}\right) \in \mathfrak{R}^{18\times 18}$ is the matrix of the centrifugal force and Coriolis force. $G(q) \in \mathfrak{R}^{18\times 1}$ is the matrix of the gravity term. In the dynamic equation, the acceleration, torque and contact force are defined as state variables. The advantage is that one or more of the state variables can be directly restricted and controlled. As an example, it is feasible to control the contact force, or switch to the torque output mode or position output mode, which makes the control expression more concise.

The actual COM position of the robot changes during walking. The actual COM position can be calculated in real-time according to the actual joint position and mass parameters of the robot. The COM is calculated as:

$$P_{com}(q) = \frac{\sum_{i=1}^{n} m_i p_i(q)}{\sum_{i=1}^{n} m_i}, \tag{12}$$

where m_i is the mass of link i and p_i is the COM position of link i in the world coordinate frame.

During walking, it is necessary to track the position and posture of each foot, the posture of the upper body, the position of each arm and one or more joint positions and perform other tasks. The position in the world coordinate frame is; therefore, eliminated and the relative position between the COM and foot is controlled. All tasks are described by:

$$T(q) = \begin{bmatrix} P_{com \to fr} & P_{com \to fl} & R_b & R_{fr} & R_{fl} \end{bmatrix}^T, \tag{13}$$

In Equation (13), $P_{com \to fr}$ and $P_{com \to fl}$ are the relative positions of the foot and COM in the world coordinate frame, R_{fr} and R_{fl} are the attitude of the foot in the world coordinate frame, R_b is the posture of the upper body in the world coordinate frame, and the posture is expressed by the roll, pitch and yaw.

3.2. Task Space Controller

In the task space, we only care about the kinematics part of the task space performance, ignoring the dynamics of the robot. A feedforward and a proportional and derivative feedback controller are applied to control the task acceleration. Through configuration of the gains k_d and k_p, the error state can approach the zero equilibrium point. Although different control targets are unified as a task in the task space, the weight of each control task depends on the scenario. In a walking scenario, the tracking of leg dynamics should be good, whereas the error in tracking the upper body and arms only needs to be not too large. When performing working tasks, the body should be flexible and the arm tracking is expected to be good. Setting different parameters for different application scenarios achieves different control effects. Equation (14) is the feedback and feedforward controller. T_d is the desired position or orientation, $\dot{T}_d$ and $\ddot{T}_d$ are its desired velocity and acceleration, respectively, as mentioned in Equation (13). $\ddot{T}_d^*$ is the newly calculated acceleration.

$$\ddot{T}_d^* = k_p(T_d - T) + k_p\left(\dot{T}_d - \dot{T}\right) + \ddot{T}_d, \tag{14}$$

In the support phase, to ensure that the foot is in contact with the ground without slippage and rotation, the velocity and acceleration of the contact point in the world coordinate system are theoretically required to be zero. For control stability, the contact constraint here is set as a soft constraint. That is to say, the expected acceleration value $\ddot{P}^*_{\text{contact}}$ of the contact-related quantity in the task space is set to zero, ignoring the velocity:

$$\ddot{P}^*_{\text{contact}} = 0, \tag{15}$$

We have obtained the control input acceleration in the task space. Taking the time derivative of Equation (13) twice, the kinematic mapping Equation (12), which maps the task space to the joint space, is obtained:

$$J(q) \cdot \ddot{q} = \ddot{T}(q) - \dot{J}(q)\dot{q}, \tag{16}$$

where $J(q)$ is the task-space Jacobian matrix.

Using Equations (14)–(16), an optimization is performed to find the state input that satisfies the task-space control acceleration:

$$\underset{\ddot{q},\tau,F}{min} \| J(q_a) \cdot \ddot{q} + \dot{J}(q_a)\dot{q}_a - \ddot{T}_d^*(q_a) \|^2, \tag{17}$$

where q_a is the actual joint angle and $\ddot{q}$ is the joint acceleration after optimization.

3.3. ZMP and Force Mapping

Given the contact force and torque, the location of the ZMP in the foot coordinate frame is calculated using Equation (18). The force and torque are expressed in the local coordinate frame of the foot as:

$$p = \begin{bmatrix} -{}^{b}M_y/{}^{b}F_z \\ {}^{b}M_x/{}^{b}F_z \end{bmatrix}, \tag{18}$$

A contact force term is introduced to the optimization goal to ensure the stability of the ground contact force and the stability of the solver when the robot is disturbed or walking on uneven ground. The control input mapping by optimization is Equation (19). The F_{ref} is the reference force, and the F is the contact force calculated by model dynamics and control input instead of measured by the force sensor. Equation (19) stabilizes ZMP by controlling instantaneous dynamics of the whole body, while Equation (9) stabilizes ZMP by adjusting the COM trajectory. The two controllers are combined to achieve a wider range of stability control.

$$\underset{\ddot{q},\tau,F}{min} \|p_x^* \cdot {}^{b}F_z + {}^{b}M_y\|^2 + \|p_y^* \cdot {}^{b}F_z - {}^{b}M_x\|^2 + \|F - F_{ref}\|^2. \tag{19}$$

3.4. Constraints

When the robot is walking at high speed, the swinging leg rapidly moves in the forward direction, generating a rotational moment in the vertical direction of the supporting foot. This destabilizes the foot–surface contact of the supporting leg and rotates the robot. In addition, motion planning does not consider whether the ground friction meets requirements. Without limits, the robot foot slips on the ground, making the robot unstable, as shown in Figure 5. With our improved dynamic equations, the required frictional force and torque can be adopted as constraints.

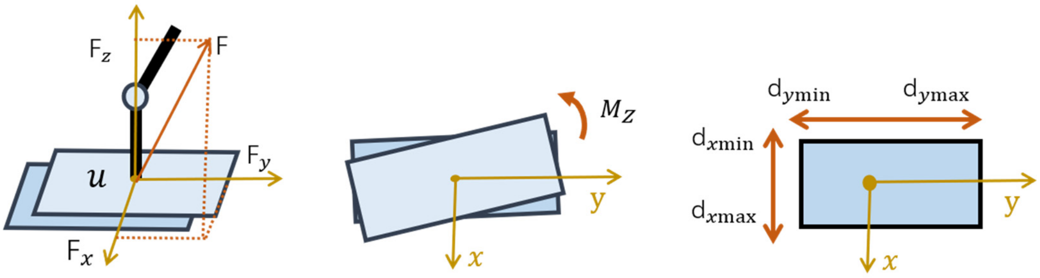

Figure 5. Friction and support region constraints.

The coefficient of ground friction is denoted as u. The contact force needs to meet the conditions given in Equation (20) to prevent slipping. We also added the ZMP constraint to ensure the ZMP is in the support area. The conditions are:

$$\begin{cases} |F_x| - uF_z \le 0 \\ \left|F_y\right| - uF_z \le 0 \\ |M_z| < M_{max} \end{cases} \tag{20}$$

It is necessary to constrain the amplitudes of the control variables in order to avoid joint actuator saturation. Boundary constraints of the acceleration amplitude and moment amplitude can be set according to actual robot hardware conditions:

$$lb \le X \le ub. \tag{21}$$

3.5. Prioritized Whole-Body Mapping

The task space controller and ZMP controller are often in conflict with each other. We weight the command tasks to select appropriate whole-body joint torques according to task importance through convex optimization. The dynamic equation is used as the constraint equation. The final optimization form is:

$$\begin{array}{ll} \min\limits_{\ddot{q},\tau,F} & \frac{1}{2}w^2{}_{task}\|\mathrm{J}(q)\ddot{q}+\dot{\mathrm{J}}(q)\dot{q}-\ddot{\mathrm{T}}^{*}_{d}(q)\|^{2}+\sum\frac{1}{2}w^2{}_{zmp}\|\mathrm{p}_x\cdot{}^{b}F_z+{}^{b}M_y\|^2+\frac{1}{2}w^2{}_{F}\|F-F_{ref}\|^2 \\ s.t. & \begin{bmatrix} D(q) & -B & -J^T(q) \end{bmatrix}\cdot\mathrm{X}+\mathrm{C}\left(q,\dot{q}\right)\dot{q}+G(q)=0 \\ & \mathrm{d}_{x\min}\le -{}^{b}M_y/{}^{b}F_z\le \mathrm{d}_{x\max} \\ & \mathrm{d}_{y\min}\le {}^{b}M_x/{}^{b}F_z\le \mathrm{d}_{y\max} \\ & |\mathrm{F}_x|-u\mathrm{F}_z\le 0 \\ & \left|\mathrm{F}_y\right|-u\mathrm{F}_z\le 0 \\ & lb\le \mathrm{X}\le ub \end{array} \tag{22}$$

where $\mathrm{w}_{task}=\begin{bmatrix} w_{Pcom\rightarrow\mathrm{foot}} & w_{Pcom\rightarrow foot} & w_{Rb} & w_{\mathrm{foot}} & w_{foot} \end{bmatrix}$ is the weight coefficient of each task in the task space, $\mathrm{w}_{\mathrm{zmp}}$ is the ZMP tracking weight and w_F is the contact force weight. Among walking tasks, tracking of the COM trajectory is set to have a high priority and other tasks are set to have a low priority. The general form of quadratic programming is shown as Equation (23), rewrite the general form cost function as $0.5\|A\chi-b\|^2$, thus $G=A^TA$ and $g=-A^Tb$. A and b can be decomposed into smaller blocks as Equation (24). Using this rewrite method, the optimization problem (22) can be rewritten into a general form and solved by a standard solver.

$$\begin{array}{ll} \underset{\chi}{min} & 0.5\chi^TG\chi+g^T\ \chi \\ & A_{ueq}\chi\le b_{ueq} \\ st. & A_{eq}\chi=b_{eq} \\ & lb\le\chi\le ub \end{array} \tag{23}$$

$$A=\begin{bmatrix} \omega_0A_0 \\ \omega_1A_1 \\ \cdot \\ \omega_nA_n \end{bmatrix}, b=\begin{bmatrix} \omega_0b_0 \\ \omega_1b_1 \\ \cdot \\ \omega_nb_n \end{bmatrix} \tag{24}$$

The above optimization problem is complex but can be solved by the *MATLAB* QP solver quadprog in 1 ms. The feedback parameters and weight parameters of the method have a significant impact on the results. Although there are many parameters involved in this paper, they all correspond to relatively clear physical meanings. High feedback gain has better tracking effect and low feedback gain has better flexibility. When debugging the parameters, we can first plan a squat down and stand up movement, and set the feedback gain to be larger when moving without disturbance. The weight coefficient is designed according to the priority of task importance to get rough parameters. At high speed, the parameters are adjusted in detail. Generally, the feedback gain is reduced to improve the flexibility. Generally, after one or two days of adjustment the controller has a good performance, and when the dynamic model parameters change, only fine tuning is needed.

4. Simulation and Results

We used MATLAB2018 and Simscape [19] software to verify the methods mentioned above. Simscape is a toolbox of MATLAB for physics modelling, as a package of Simulink with tools for modelling. As shown in Figure 6, we first designed the mechanical structure of the humanoid robot in SolidWorks and then divided the robot into multiple links with the joint as the segmentation point. In MATLAB the joints are added between the separate links, and the joints take torque as input and angle as output. At the same time, six degrees of freedom free movement joints were added between the body and the world, and their states were measured as inertial measurement unit (IMU) sensor output. For contact, solid-to-solid is not directly used for foot contact. Instead, four balls are added to the four endpoints of the foot to construct a sphere-to-solid contact. The advantage is that the computation is fast and stable, and the SM_Contact_Forces_Lib_R18a_v4p1 contact library is used for contact design in the simulation. To make the simulation more realistic, gaussian noise and low pass filter are added to the state value, including joint velocity, IMU data and contact force.

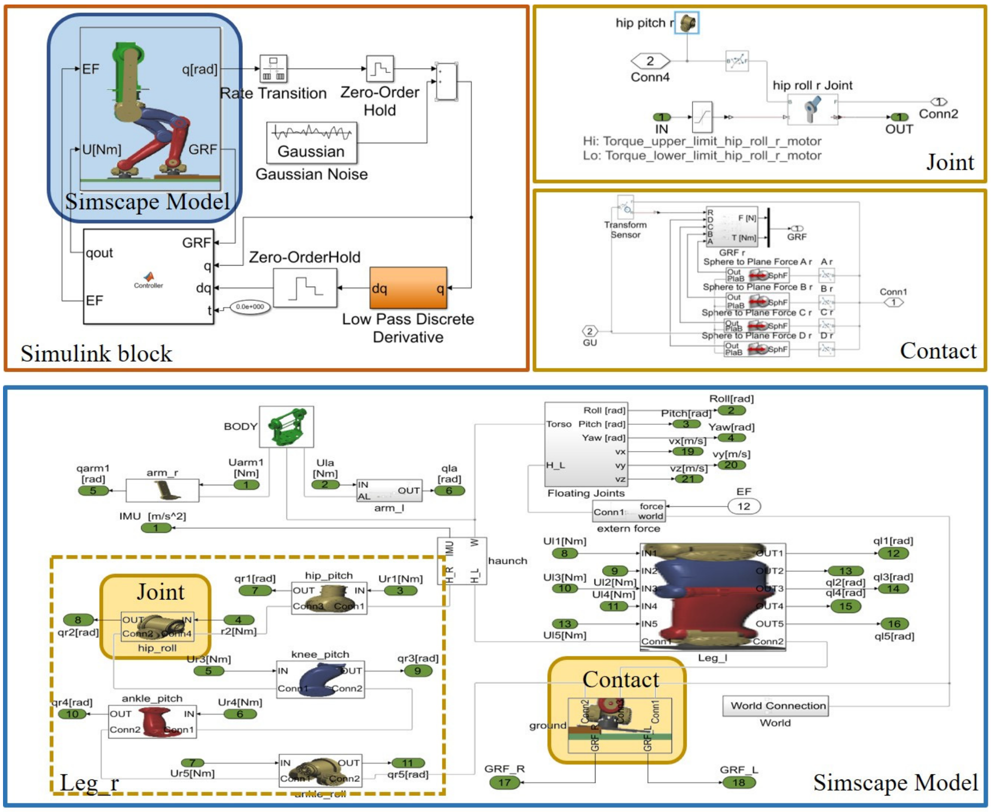

Figure 6. Simscape block of the simulated model.

The simulation is generated in three stages: The first is to verify external force disturbance, second is to verify the unmodeled dynamics disturbance in fast waking and the last is to verify uneven ground disturbance. The total mass of the model is 40.58 kg; the mass of a single leg is about 10.19 kg, the mass of a single arm is 3.1 kg. Table 1 shows the mass and inertia of the links in the model, and also lists the dimensions of the different links of the robotic mechanism. The corresponding variables are labeled in Figure 4b. The coefficient of friction between synthetic rubber and concrete is 0.6–0.85, so the friction coefficient is set to 0.7. The robot has five DOFs on each leg (i.e., two on the hip, one on the knee and two on the ankle) and one DOF on the shoulder in the pitch direction on each arm. Table 2 gives the weights and feedback gains of the task space controller. The weights and feedback gain in Table 1 can be adjusted manually to achieve different control effects.

Table 1. Model parameters in simulation.

Description	Value
body mass and inertia	14 kg, [0.82,0.58,0.28] × 2 kg·m^2
arm mass and inertia	3.10 × 2 kg, [0.006,0.133,0.133] × 2 kg·m^2
hip mass and inertia	2.05 × 2 kg, [0.003.0.002,0.004] × 2 kg·m^2
thigh mass and inertia	3.15 × 2 kg, [0.017,0.003,0.018] × 2 kg·m^2
shank mass and inertia	3.15 × 2 kg, [0.002,0.037,0.036] × 2 kg·m^2
ankle mass and inertia	0.94 × 2 kg, [0.002,0.001,0.002] × 2 kg·m^2
foot mass and inertia	0.90 × 2 kg, [0.002,0.002,0.0008] × 2 kg·m^2
L_{arm}	0.4 m
$L_{shoulder}$	0.504 m
L_{body}	0.459 m
W_{hip}	0.16 m
L_{hip}	0.056 m
L_{thigh}	0.274 m
L_{shank}	0.33 m
L_{ankle}	0.05 m
H_{foot}	0.061 m
L_{foot}	0.269 m
W_{foot}	0.16 m

Table 2. Weights and gains in simulation.

Task	Weight	Kp	Kd
$P_{com \rightarrow foot}$	1	[1000,1000,700]	[40,40,65]
R_b	0.6	[10,50,50]	[3.2,5.5,5.5]
R_{fr}	1	[150,150,90]	[33.9.33.9,24]
$R_{\rm arm}$	0.5	[50,50]	[5.5,5.5]
ZMP	0.003	[40,40]	[3.2,3.2]
F	0.001	-	-

4.1. Disturbance Recovery in Walking

In verifying the robot's ability to recover from disturbances in the walking process, we first used a preview control to generate the COM reference trajectory and reference contact force. Given a sequence of 10 foot placements (i.e., 10 steps) with a velocity of 1.44 km/h, the duration of each step is 0.5 s, the step length is 0.2 m, and the period of double support accounts for 5% of the step cycle. As shown in Figure 7, at t = 1.4 s, the left leg of the robot is the supporting leg and the right leg is the swinging leg. An external disturbance is applied to the waist of the robot in the forward, right and vertical directions with magnitudes of 400, 20 and 300 N, respectively. At t = 1.5 s, the disturbance is cancelled (i.e., the disturbance lasts 0.1 s). The robot gains an impulse of 40, 2 and 30 NM in the three directions, respectively. When impacted by external forces, the robot generates additional movement in the direction of the impact to maintain balance. The simulated motion is shown in the supplemental videos (In Supplementary).

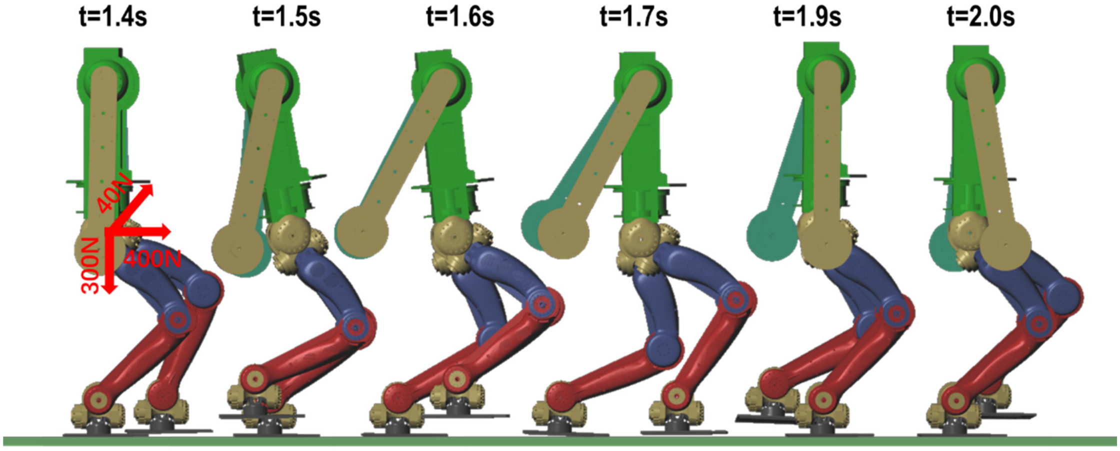

Figure 7. Disturbance recovery in walking.

Figure 8 shows the COM tracking error and body posture error. At t = 1.4 s, the posture error of the upper body starts to increase. The error reaches a maximum value, about 0.08 rad, at t = 1.52 s and then gradually decreases. The pitch direction returns to the expected value at t = 1.75 s. The posture does not change appreciably in the roll direction upon impact. One reason is that the given impact is small. Another reason is that the speed of the swinging leg in the lateral direction is low during walking. The controller exerts more influence on attitude control. Throughout the walking process, there is a maximum posture error of 0.005 rad in the yaw direction, which is not caused by the rotation or slip of the robot's foot. This is because the robot model used in this article lacks hip yaw joints, and the pitch joints and roll joints at the hip and ankle are not coplanar. The inverse kinematics are over constrained. It is not possible to guarantee all postures and positions of the foot relative to the body at the same time. Instead, we introduce weights into the calculation. The maximum errors in the location of the COM in the three directions are 0.03, 0.025 and 0.09 m. At t = 1.799 s, the y-direction error reaches a maximum. At t = 2 s, the robot returns to normal walking. When the disturbance disappears, the posture and position error decreases immediately. This is because the speed error generated by the impact is large. To ensure the robot ZMP is located in the foot plate, the controller optimization solution favors stability over COM and posture tracking. During the period t = 2 to 6 s, the robot walks free of disturbance. The error reaches a maximum of 0.02 m. In fact, it is possible to configure a higher gain of the feedback controller to reduce the error, but this is unnecessary because the appropriate gains ensure both tracking and compliance performance.

Figure 9 shows the ZMP tracking result. When the robot is disturbed by the impact, owing to the compliance provided by the control method, the robot ZMP does not suddenly change, causing the robot's foot to rotate away from the ground. With the controller, the ZMP slowly changes along the direction of the impact force. When the impact disappears, the track of the actual ZMP gradually approaches the rack of the expected ZMP. Throughout the walking process, the ZMP is located in the stable support area, achieving stable control. The ability to reject 400 N and last 0.1 s external force in this paper is better than literature [20], which is 350 N and 0.05 s, respectively, and it also better than literature [21], which is 6.0 Nm. This simulation proves that when the model is disturbed by external forces, it can quickly recover the original stable walking state and achieve the desired goal.

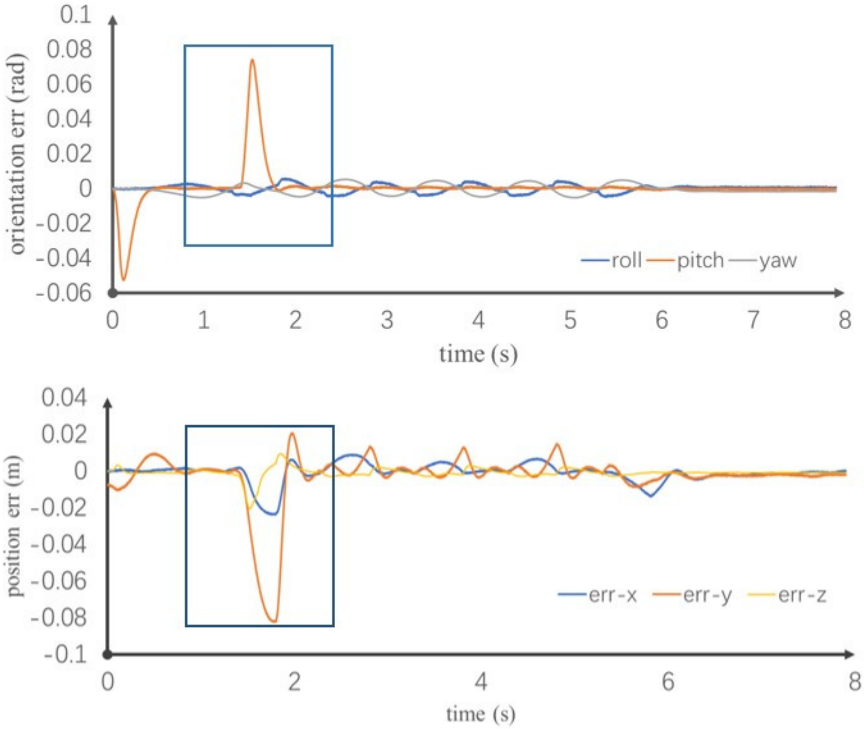

Figure 8. Center of mass (COM) tracking error and body posture error.

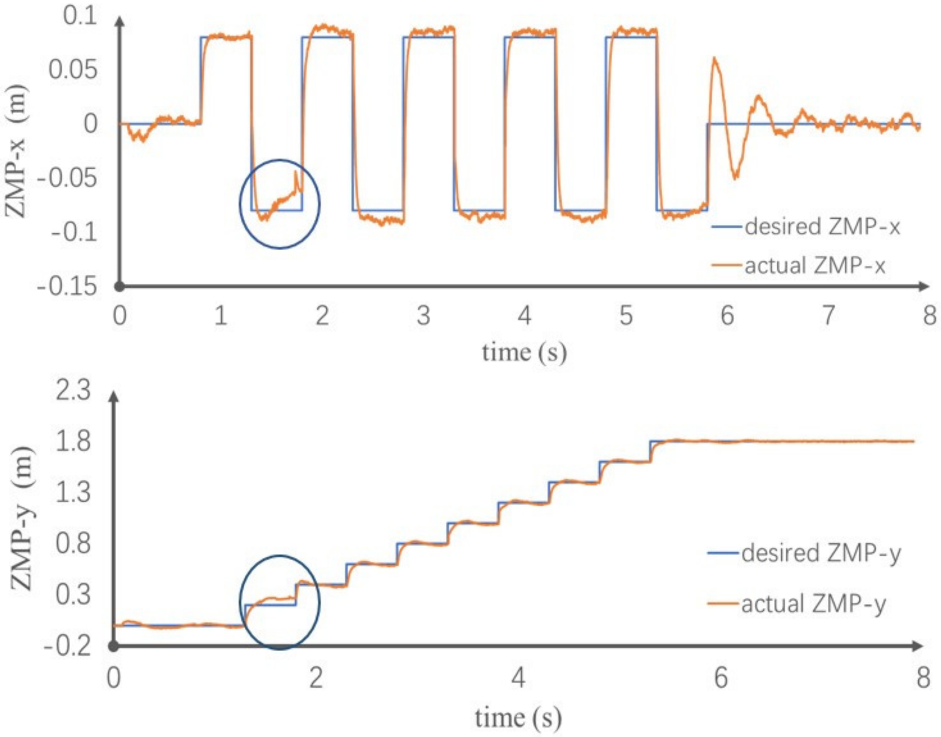

Figure 9. Zero moment point (ZMP) tracking in x and y directions.

4.2. Fast Walking

We use the proposed controller to achieve fast walking with a velocity of 6 km/h, including acceleration and deceleration gait. Similar to the previous simulation, the step length is set at 0.5 m, the step cycle is set at 0.3 s and there is no period of biped support. Meanwhile, to verify the control performance and model parameters error, we randomly adjust the mass and inertia so that it has an error of plus or minus 15%. Figure 10 shows the COM speed in the forward direction during walking. The robot reaches a maximum speed of 6.2 km/h at t = 1.5 s, which is slightly higher than the intended 6 km/h. The tracking is more accurate when the robot speed is low. During high-speed walking, when the support phase changes, the swinging leg strikes the ground and there is a sudden change in speed. For walking at high speeds, especially with large leg inertia, the rotation torque provided by the motion of the swinging leg is large, which generates a large rotation moment at the support foot. By limiting the contact torque of the foot in this article, the rotation torque of the robot is reduced to 10 Nm. Through this simulation, we prove that the method in this paper can effectively solve the problem of supporting foot rotation caused by the fast swing of the swinging leg when walking at high speed, and control the rotation torque within the limit of friction. In addition, even though the model parameter is not accurate, it can show good robustness.

Fast walking with different friction is also simulated. We changed the set of friction coefficient in the simulation environment and friction constraint of optimization. We tested different friction coefficients with 0.7, 0.4, 0.3, 0.27 and 0.25 in 6 km/h fast walking. Result shows that in 6 km/h walking, the proposed method is able to achieve a minimum friction coefficient of 0.25 walking. The simulated motion is shown in the supplemental video.

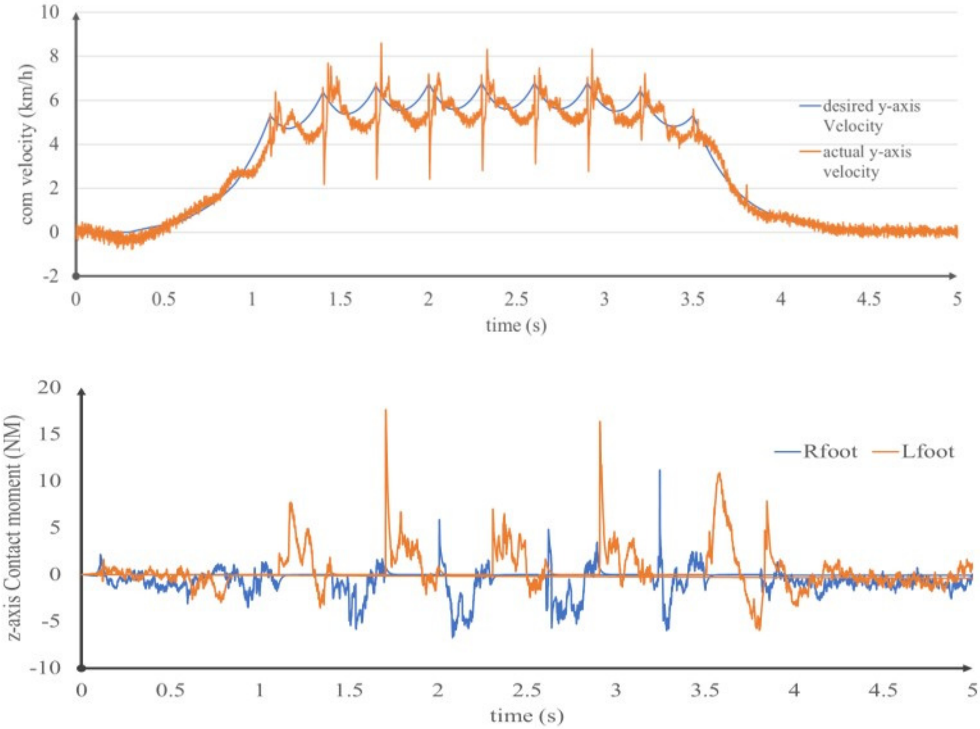

Figure 10. Center of mass (COM) velocity and contact moment in walking at 6 km/h.

4.3. Uneven Ground Walking

The control method proposed in this paper also has good adaptability for robots walking on unknown uneven ground, which included stairs of unknown size and dynamic slopes of unknown angle. In simulation, the robot does not know where the stairs or slope are, nor does it know the width or height of the stairs and angle of the slope. In order to verify the control performance of the proposed method on the unknown uneven ground, the ground is assumed to be horizontal in motion planning. A 1.44 km/h walking trajectory is planned, and the uneven ground is taken as the disturbance for real-time control. Figure 11a,b show two stairs walking up and two stairs walking down, each stair has a height of 4 cm. Figure 11c,d show the robot walking up and down the slope with a slope angle of 10°. To verify the robot's ability of walking on dynamic ground, we applied a sine motion with an amplitude of 5° and a period of 2.5 s to the ground. As shown in Figure 12, during the sinusoidal movement of the ground, the measured roll and pitch angles were lower than 0.5°, and the robot was walking steadily. The simulated motion is shown in the supplemental video.

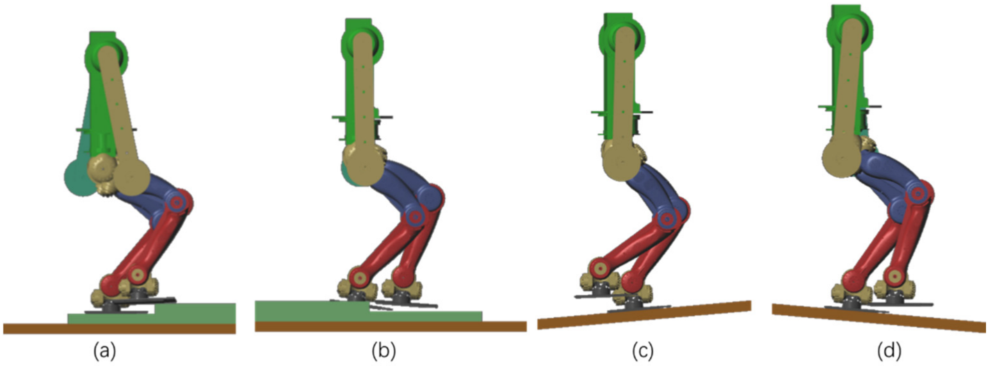

Figure 11. Walking on uneven ground: (**a**) Walking up stairs; (**b**) walking down stairs; (**c**) walking up slope; (**d**) walking down slope.

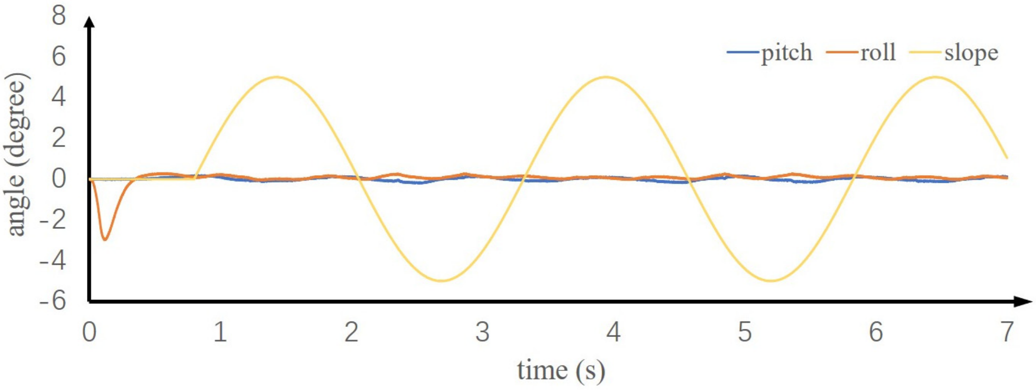

Figure 12. The angle of the body in the direction of pitch and roll while walking on the dynamic ground.

5. Discussion

In this paper, the method based on whole-body QP optimization is applied to achieve disturbance recovery. However, it can be predicted that the whole-body optimization using sequential quadratic programming (SQP) or MPC can obtain better control effect, but the calculation cost will be significantly increased, which is difficult to realize under the current calculation conditions. On the other hand, although it is not the research content of this paper, the combination of real-time planning of high-level trajectory based on state feedback, such as recalculating the foot placement and changing the trajectory of the COM, can achieve a larger range of disturbance control. For example, a three-dimensional spring-involved inverted pendulum model and a dead-beat controller could be used to re-plan the trajectory by adjusting the foothold [22], or a real-time gait generation method based on an analytical solution could be used for trajectory adjustment [23]. Despite all above advantages, this controller is implemented only in simulation. Implementing on the experimental setup has more practical challenges. For example, accurate state estimation to obtain the robot state, especially the velocity, and accurate joint force control are some of the main challenges of experimental implementation that will be discussed in the future works.

6. Conclusions

In this paper, we used a two-level controller, of MPC controller and whole-body controller, to realize generalized disturbance rejection for a biped model and get a better performance in push recovery, uneven ground walking and fast walking compared to a simplified model. The model can cover a 400 N external force, 4 cm uneven stairs and 10° slope without replanning the trajectory. Besides, this method can also control the model waking at a velocity of 6 km/h. The most important contribution of this paper is that we combine the simplified model and the whole-body model controller. Firstly, we use MPC to improve the ZMP tracking based on the simplified model. Secondly, the robot joint torque, joint acceleration and contact force are constructed as state variables, which facilitates the direct control of the included state variables. Thirdly, the task space controller is constructed, the mapping of the task space to the joint space control mode is realized through optimization, and whole-body motion is used to improve the walking stability. Finally, the ground friction constraint is added to solve the problem of a foot slipping and rotating in high-speed walking. The control framework used in this paper has good extensibility, in that it is easy to add or remove control targets to achieve more tasks.

Supplementary Materials: The following are available online at http://www.mdpi.com/1424-8220/20/10/2971/s1, Video S1: Simulation environment; Video S2: Simulation of disturbance recovery based on MPC and whole-body dynamics control of biped walking.

Author Contributions: Conceptualization, X.S. and Y.L. (Yizhou Lu); formal analysis, X.S. and D.T.; funding acquisition, J.G.; methodology, X.S. and Y.L. (Yizhou Lu); resources, D.T.; software, X.S. and Y.L. (Yi Liu); writing—original draft, X.S.; writing—review and editing, X.S. and J.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under grant number 611175077, the National Research Project under grant number B2220132014 and the National High-Tech R&D Program of China under grant number 2015AA042201.

Acknowledgments: The authors appreciate guidance from Qiang Huang and help from other colleagues at the Beijing Advanced Innovation Center for Intelligent Robots and Systems. The authors thank Glenn Pennycook, from Liwen Bianji, Edanz Group China (www.liwenbianji.cn/ac), for editing the English text of a draft of this manuscript.

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

References

1. Kajita, S.; Hirukawa, H.; Harada, K.; Yokoi, K. *Introduction to Humanoid Robotics*; Springer: Berlin/Heidelberg, Germany, 2014; Volume 101, p. 2014.
2. Vukobratovi, M.; Borovac, B. Zero-moment point—Thirty five years of its life. *Int. J. Hum. Robot.* **2004**, *1*, 157–173. [CrossRef]
3. Ott, C.; Roa, M.A.; Hirzinger, G. Posture and balance control for biped robots based on contact force optimization. In Proceedings of the 2011 11th IEEE-RAS International Conference on Humanoid Robots, Bled, Slovenia, 26–28 October 2011; pp. 26–33.
4. Kasaei, M.; Lau, N.; Pereira, A. A Hierarchical Framework to Generate Robust Biped Locomotion Based on Divergent Component of Motion. *arXiv* **2019**, arXiv:1911.07505.
5. Kajita, S.; Morisawa, M.; Miura, K.; Nakaoka, S.I.; Harada, K.; Kaneko, K.; Yokoi, K. Biped walking stabilization based on linear inverted pendulum tracking. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; pp. 4489–4496.
6. Furusho, J.; Sano, A. Sensor-based control of a nine-link biped. *Int. J. Robot. Res.* **1990**, *9*, 83–98. [CrossRef]
7. Nagasaka, K.I. Stabilization of dynamic walk on a humanoid using torso position compliance control. In Proceedings of the 17th Annual Conference on Robotics Society of Japan, Tokyo, Japan, 3–5 September 1999.
8. Kagami, S.; Nishiwaki, K.; Sugihara, T.; Kuffner, J.J.; Inaba, M.; Inoue, H. Design and implementation of software research platform for humanoid robotics: H6. In Proceedings of the 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164), Seoul, Korea, 21–26 May 2001; pp. 2431–2436.
9. Hirai, K.; Hirose, M.; Haikawa, Y.; Takenaka, T. The development of Honda humanoid robot. In Proceedings of the 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146), Leuven, Belgium, 20–22 May 1998; pp. 1321–1326.
10. Yu, Z.; Zhou, Q.; Chen, X.; Li, Q.; Meng, L.; Zhang, W.; Huang, Q. Disturbance rejection for biped walking using zero-moment point variation based on body acceleration. *IEEE Trans. Ind. Inform.* **2018**, *15*, 2265–2276. [CrossRef]

11. Li, Z.; Zhou, C.; Zhu, Q.; Xiong, R.; Caldwell, D. Active Control of Under-actuated Foot Tilting for Humanoid Push Recovery. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; IEEE: Piscataway, NJ, USA, 2015.
12. Vukobratovic, M.; Borovac, B.; Surla, D.; Stokic, D. *Biped Locomotion: Dynamics, Stability, Control and Application*; Springer Science & Business Media: Berlin, Germany, 2012; Volume 7.
13. Yoshino, R. Stabilizing control of high-speed walking robot by walking pattern regulator. *J. Robot. Soc. Jpn.* **2000**, *18*, 1122–1132. [CrossRef]
14. Khatib, O. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE J. Robot. Autom.* **1987**, *3*, 43–53. [CrossRef]
15. Hutter, M.; Hoepflinger, M.A.; Gehring, C.; Bloesch, M.; Remy, C.D.; Siegwart, R. Hybrid operational space control for compliant legged systems. *Robotics* **2013**, 129.
16. Caron, S.; Kheddar, A.; Tempier, O. Stair climbing stabilization of the HRP-4 humanoid robot using whole-body admittance control. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 277–283.
17. Feng, S.; Whitman, E.; Xinjilefu, X.; Atkeson, C.G. Optimization-based full body control for the darpa robotics challenge. *J. Field Robot.* **2015**, *32*, 293–312. [CrossRef]
18. Deng, H.; Ohtsuka, T. A Parallel Code Generation Toolkit for Nonlinear Model Predictive Control. Available online: https://github.com/deng-haoyang/ParNMPC (accessed on 5 October 2019).
19. MathWorks. Simscape ™ Multibody ™ User's Guide, © COPYRIGHT 2002–2019 by The MathWorks. Available online: https://ww2.mathworks.cn/products/simscape.html (accessed on 5 August 2019).
20. Shafiee, M.; Romualdi, G.; Dafarra, S.; Chavez, F.J.A.; Pucci, D. Online DCM Trajectory Generation for Push Recovery of Torque-Controlled Humanoid Robots. *arXiv* **2019**, arXiv:1909.10403.
21. Caron, S. Biped Stabilization by Linear Feedback of the Variable-Height Inverted Pendulum Model. *arXiv* **2019**, arXiv:1909.07732.
22. Liu, Y.; Wensing, P.M.; Orin, D.E.; Zheng, Y.F. Dynamic walking in a humanoid robot based on a 3D actuated Dual-SLIP model. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 5710–5717.
23. Tedrake, R.; Kuindersma, S.; Deits, R.; Miura, K. A closed-form solution for real-time ZMP gait generation and feedback stabilization. In Proceedings of the 2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids), Seoul, Korea, 3–5 November 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 936–940.