Robotica (2014) volume 32, pp. 467–476. © Cambridge University Press 2013
doi:10.1017/S0263574713000829

Robust push recovery by whole-body dynamics control with extremal accelerations

Xuechao Chen†‡§¶, Qiang Huang†‡§, Zhangguo Yu†‡§ and Yuepin Lu‖*

†*The Intelligent Robotics Institute, School of Mechatronical Engineering, Beijing Institute of Technology, Beijing, P. R. China*
‡*Key Laboratory of Biomimetic Robots and Systems, Ministry of Education, Beijing, P. R. China*
§*Key Laboratory of Intelligent Control and Decision of Complex System, Beijing, P. R. China*
¶*The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA*
‖*School of Mechanical Engineering, University of Science and Technology Beijing, Beijing, P. R. China*

(Accepted July 31, 2013. First published online: August 28, 2013)

SUMMARY
This paper presents a whole-body dynamics controller for robust push recovery on a force-controlled bipedal robot. Featherstone's spatial vector method is used to deduce dynamics formulas. We reveal a relationship between the accelerations of the floating base and the desired external forces needed for those accelerations. Introducing constraints on the desired external forces causes corresponding constraints on the accelerations. Quadratic programming is applied to find the extremal accelerations, which recover the robot from pushes as best as possible. A robustness criterion is proposed based on the linear inverted pendulum model to evaluate the performance of push recovery methods quantitatively. We evaluate four typical push recovery methods and the results show that our method is more robust than these. The effectiveness of the proposed method is demonstrated by push recovery in simulations.

KEYWORDS: Push recovery; Whole-body dynamics; Bipedal robot; Force control; Quadratic programming.

1. Introduction

Force-controlled bipedal robots have become more popular due to increasing requirements for the robot,[1–4] such as safety in the human–robot interaction and robustness to unknown disturbances. Push, as one of the most common physical interactions between human beings, is the most popular example for studying disturbance recovery. Generally, a push, which acts on a robot for a certain time, causes the robot to have an initial velocity. Researchers have proposed many methods that allow robots to recover their original positions.

Model predictive control (MPC) can generate new motions for recovery from small perturbations.[5–7] This method solved a quadratic programming (QP) problem on a simplified linear inverted pendulum model (LIPM), which constrained the center of mass (CoM) at a constant height. Unfortunately, it was relatively conservative and did not take advantage of the whole-body motion for recovery, which was why it was only adequate for small disturbances.

Linear quadratic regulators were used to generate controllers for different perturbations.[8] This method needed to change the optimization criterion according to the size of the perturbation. In order to treat different perturbations uniformly, one optimization criterion was used to generate multiple balance strategies for various impulsive perturbations.[9] Later, this method was extended to solve both impulsive and constant push problems in ref. [10]. To improve the robustness of the robot

* Corresponding author. E-mail: luyuepin@126.com.

for disturbances, differential dynamic programming (DDP) was employed to generate an optimal trajectory library for standing balance.[11] These methods used multi-link models, which were more accurate than the model used by MPC. However, they only considered the boundaries of joint torques and joint ranges rather than the constraints on the external forces, such as zero-moment point (ZMP), friction and unilateral vertical force, which had two potential problems. On the one hand, those boundaries cannot guarantee that the desired external forces are equal to the actual ones (ground reaction forces, GRFs), which means if the desired external forces are not equal to the GRFs, the robot would experience unexpected foot rotation, slipping or tipping over. On the other hand, these methods cannot enable the robot to recover from perturbations as best as possible because they do not use the extremal accelerations, which are on the boundary of the acceleration inequalities.

In this paper, we will propose a more robust method, which uses whole-body dynamics for push recovery by employing extremal accelerations. We study the LIPM-based method for push recovery and propose a criterion to evaluate the robustness of push recovery methods. The spatial vector method is used to deduce dynamics formulas and a relationship between the accelerations of the floating base and the desired external forces is revealed. The desired external forces are constrained by ZMP, friction and unilateral vertical forces, which causes corresponding constraints on the accelerations. Quadratic programming is applied to find the extremal accelerations, which enable the robot to recover from pushes as best as possible. The results show that the robustness of our method is greater than the presented methods.

The rest of this paper is organized as follows. Section 2 will introduce the robustness criterion. Section 3 will present the spatial method to calculate inverse dynamics. The push recovery controller will be presented in Section 4. Section 5 will present how to obtain extremal accelerations. The simulations and conclusions will be provided in Sections 6 and 7, respectively.

2. Robustness Criterion

Up to now, researchers have seldom proposed a means to evaluate the robustness of push recovery methods. Most researchers used only the biggest impulse, divided by the total mass, as a criterion. However, these two parameters restrict a consideration of robustness because the size of the feet and the height of CoM, which obviously influence robustness, were not part of the equation. Researchers used different models, such as the two-link model, the three-link model or the full-body model. Obviously, different models affect robustness evaluation. However, in order to establish a uniform criterion to evaluate the robustness of push recovery methods, it would be better to use the basic effect factors on the robustness evaluation. So the simplest model, LIPM, is used to establish the criterion.

In ref. [12], when the robot, which was simplified as an LIPM, recovered from the biggest disturbance, it applied its extremal acceleration to stop the CoM first and then move the CoM back to the original position. When the ZMP is located in the toe, the acceleration is the extremal one, which represents the best recovery capability. Therefore, when we establish the criterion, we should evaluate this best recovery capability.

The threshold condition is that the CoM stops at the upright of the toe. According to the orbital energy theory of the LIPM,[13] we obtain

$$\frac{1}{2}V_0^2 - \frac{g}{2H_{\mathrm{CoM}}}L_{\mathrm{ff}}^2 = 0, \tag{1}$$

where H_{CoM} is the height of the CoM, g is the gravitational acceleration, V_0 is the velocity after the CoM suffers from a push, and L_{ff} is the length of the forefoot. In this case, V_0 equals the maximum velocity $V_{\max}$ that the robot can recover from. So

$$V_{\max} = V_0 = \sqrt{\frac{g}{H_{\mathrm{CoM}}}}L_{\mathrm{ff}}. \tag{2}$$

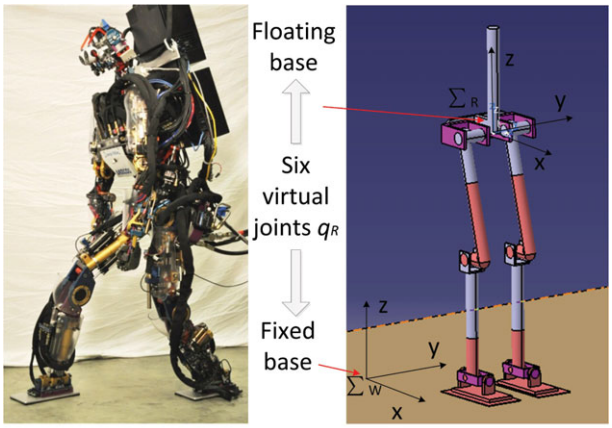
According to the impulse–momentum theory,

$$V_{\max} = \frac{I_{\max}}{M}, \tag{3}$$

https://doi.org/10.1017/S0263574713000829 Published online by Cambridge University Press

Table I. Robustness comparison.

Reference	5	9	10	11
I (N s)	22	25	19	25
H_{CoM} (m)	0.95	1	0.75	0.65
M (kg)	96.9	70	48.97	48.97
L_{ff} (m)	0.18	0.1	0.1	0.1
Model	LIPM	4 links	4 links	2 links
K_{robust}	0.39	1.14	1.07	1.31



(a) (b)

Fig. 1. (Colour online) The Sarcos Primus humanoid robot (a) and its simplified model (b).

where $I_{\max}$ is the impulse of the biggest push that the robot can stand and M is the total mass of the robot.

For normalization, the robustness criterion is found as

$$K_{\mathrm{robust}} = \frac{I_{\max} * T_C}{M * L_{\mathrm{ff}}}, \tag{4}$$

where $T_c = \sqrt{H_{\mathrm{CoM}}/g}$. For the push recovery methods based on LIPM, their maximum robustness value is 1.

Using (4), we compare the robustness of four typical methods, which is shown in Table I.

3. Whole-Body Dynamics Formulation

The Sarcos Primus humanoid robot, as shown in Fig. 1(a), is a hydraulic force-controlled humanoid robot. When building its model in the simulator (see Fig. 1b), its upper body is simplified to focus on its integrated mass–inertia properties. The Young's modulus–coefficient of restitution element is used instead of the spring–damper model as the contact model between the feet and the ground in the simulator.[14] There are seven joints in each leg: two in the hip, one in the thigh, one in the knee, one in the shank and two in the ankle. The thigh joint is fixed, so $q_{\mathrm{rl}} \in R^{6\times1}$ and $q_{\mathrm{ll}} \in R^{6\times1}$ represent the angles of the other six joints of the right leg and left leg, respectively. Featherstone's spatial vector method,[15,16] which is an efficient rigid-body dynamics formulation, is used to solve our rigid-body dynamics problem. When using it, the frame, expressing the variables, must be specified. In this paper, left superscripts denote the reference frames and variables with $\hat{}$ superscripts denote spatial variables. The floating-base coordinate frame $\sum_R$ is located at the center of the pelvis, and the fixed-base coordinate frame $\sum_W$, also known as the world frame, is on the ground. The floating base is the root of this model, which is built using a kinematic tree. Six virtual joints are introduced between $\sum_W$ and $\sum_R$. The first three joints are translations in the x, y and z directions, and the second three are successive rotations about the x, y and z axes in that order. These virtual joints are

https://doi.org/10.1017/S0263574713000829 Published online by Cambridge University Press

used to express the location of this model in the world frame. Note that $q_r \in R^{6\times 1}$ represents the positions and angles of these joints.

Introducing kinematic constraints

$$\frac{d}{dt}\left({}^{W}J_{\mathrm{rf}}\dot{q}\right) = 0 \tag{5}$$

and/or

$$\frac{d}{dt}\left({}^{W}J_{\mathrm{lf}}\dot{q}\right) = 0, \tag{6}$$

which is due to the contact between the feet and the ground, the dynamics of this model is written as

$$M(q)\ddot{q} + C(q,\dot{q}) = \tau + {}^{W}J_{\mathrm{rf}}^{T}\,{}^{W}\hat{f}_{\mathrm{rf}} + {}^{W}J_{\mathrm{lf}}^{T}\,{}^{W}\hat{f}_{\mathrm{lf}}, \tag{7}$$

where $q = [q_r; q_{\mathrm{rl}}; q_{\mathrm{ll}}]$, $M(q) \in R^{18\times 18}$ is the joint space inertia matrix, $C(q,\dot{q}) \in R^{18\times 1}$ is a vector containing the Coriolis, centrifugal and gravitational terms, and $\tau = [0_{6\times 1}; \tau_{\mathrm{rl}}; \tau_{\mathrm{ll}}]$. Here, $\tau_{\mathrm{rl}} \in R^{6\times 1}$ and $\tau_{ll} \in R^{6\times 1}$ are torque inputs of the right leg and left leg, respectively. ${}^{W}J_{\mathrm{rf}} \in R^{6\times 18}$ is the Jacobian from the right foot coordinate frame to $\sum_W$, and ${}^{W}J_{\mathrm{lf}} \in R^{6\times 18}$ is the Jacobian from the left foot to $\sum_W$. Note that ${}^{W}\hat{f}_{\mathrm{rf}} \in R^{6\times 1}$ is the external force acting on the right foot and expressed in $\sum_W$, and ${}^{W}\hat{f}_{\mathrm{lf}} \in R^{6\times 1}$ is that of the left foot. These forces are derived from the kinematic constraints.

We will now introduce some variables, which will be used later. ${}^{W}p_r = [{}^{W}\theta_r; {}^{W}p_r] = [\theta_x, \theta_y, \theta_z, p_x, p_y, p_z]^T$ consists of the attitude and position of $\sum_B$ with respect to $\sum_W$ expressed in $\sum_W$; and ${}^{R}\hat{v}_r = [{}^{R}\omega_r; {}^{R}v_r] = [\omega_x, \omega_y, \omega_z, v_x, v_y, v_z]^T$ is the spatial velocity of the floating base expressed in $\sum_R$. ${}^{W}p_r$ is used to get the coordinate transformation matrix ${}^{R}X_W$, which transforms spatial velocity, acceleration or force from $\sum_W$ to $\sum_R$.

The spatial acceleration of the floating base ${}^{W}\hat{a}_r$ and the angular acceleration of the virtual joints are related by

$$\ddot{q}_r = {}^{W}J_R^{-1}\left({}^{W}\hat{a}_r - {}^{W}\dot{J}_R\dot{q}_r\right), \tag{8}$$

where ${}^{W}J_R$ is the Jacobian from $\sum_R$ to $\sum_W$.

With the kinematic constraints (5) and/or (6), the relationships between $\ddot{q}_{\mathrm{rl}}$, $\ddot{q}_{\mathrm{ll}}$ and $\ddot{q}_r$ are

$$\ddot{q}_{rl} = {}^{W}J_{\mathrm{rf2}}^{-1}\left(-{}^{W}\dot{J}_{\mathrm{rf}}\dot{q} - {}^{W}J_{\mathrm{rf1}}\ddot{q}_r\right), \tag{9}$$

and

$$\ddot{q}_{ll} = {}^{W}J_{\mathrm{lf3}}^{-1}\left(-{}^{W}\dot{J}_{\mathrm{lf}}\dot{q} - {}^{W}J_{\mathrm{lf1}}\ddot{q}_r\right), \tag{10}$$

respectively, where ${}^{W}J_{\mathrm{rf}} = [{}^{W}J_{\mathrm{rf1}}, {}^{W}J_{\mathrm{rf2}}, {}^{W}J_{\mathrm{rf3}}]$ and ${}^{W}J_{\mathrm{lf}} = [{}^{W}J_{\mathrm{lf1}}, {}^{W}J_{\mathrm{lf2}}, {}^{W}J_{\mathrm{lf3}}]$. ${}^{W}J_{\mathrm{rf1}}$, ${}^{W}J_{\mathrm{rf2}}$, ${}^{W}J_{\mathrm{rf3}}$, ${}^{W}J_{\mathrm{lf1}}$, ${}^{W}J_{\mathrm{lf2}}$ and ${}^{W}J_{\mathrm{lf3}} \in R^{6\times 6}$.

The relationship between spatial acceleration and conventional acceleration is

$${}^{W}\hat{a}_r = {}^{R}X_W^{-1}\left(a^c - \begin{bmatrix} 0_{3\times 1} \\ {}^{R}\omega_r \times {}^{R}v_r \end{bmatrix}\right), \tag{11}$$

where $a^c = [{}^{R}\dot{\omega}_r; {}^{R}\dot{v}_r]$ is the conventional acceleration of the floating base in $\sum_R$.

As long as the conventional acceleration of the floating base a^c is known, the angular accelerations of all joints $\ddot{q}$ can be calculated by (8)–(11). Then, joint torques can be derived by inverse dynamics.

4. Push Recovery Controller

Only contact between the feet and the ground is considered in this paper.

https://doi.org/10.1017/S0263574713000829 Published online by Cambridge University Press

4.1. Desired external forces

First, we must compute desired external forces, which can accelerate the robot with the desired accelerations. This model is treated as a fixed-base model without external forces. Then joint torques are calculated by

$$\begin{bmatrix} \tau_r \\ \tau_{\mathrm{rl}} \\ \tau_{\mathrm{ll}} \end{bmatrix} = M\ddot{q} + C, \tag{12}$$

where τ_r is the joint torques of the virtual joints. These torques result from virtual external forces acting on the floating base. Then, the desired external forces are derived through

$$^{W}\hat{f}_{\mathrm{ext}} = {}^{W}J_{R}^{-T}\tau_r, \tag{13}$$

where $^{W}\hat{f}_{\mathrm{ext}}$ is the desired total external force, which is provided by both feet.

4.2. Recovery in operational space

Proportional-Derivative (PD) control yields desired accelerations a^c which enable the robot to keep its desired positions and attitudes:

$$a^c = K_p {}^{R}R_{W} \left({}^{W}p_r^{\mathrm{ref}} - {}^{W}p_r\right) - K_d {}^{R}\hat{v}_r, \tag{14}$$

where K_p and K_d are PD gain matrices, ${}^{R}R_{W} = [I_{3\times3}\ 0_{3\times3}; 0_{3\times3}\ R^{-1}]$, R is the posture matrix of the floating base in $\sum_W$, ${}^{W}p_r^{\mathrm{ref}}$ represents desired positions and attitudes of the floating base.

Here, $\ddot{q}$ and $^{W}\hat{f}_{\mathrm{ext}}$ are obtained through (8)–(13). Then, joint torques are calculated by

$$\begin{bmatrix} 0_{6\times1} \\ \tau_{\mathrm{rl}} \\ \tau_{\mathrm{ll}} \end{bmatrix} = M\ddot{q} + C - {}^{W}J_{\mathrm{rf}}^{T}K_f {}^{W}\hat{f}_{\mathrm{ext}} - {}^{W}J_{\mathrm{lf}}^{T}(I - K_f){}^{W}\hat{f}_{\mathrm{ext}}. \tag{15}$$

The first six torques are zeros because they are virtual joints and do not have kinematic constraints on the floating base. K_f is the force distribution matrix.

5. Extremal Accelerations

Desired accelerations are obtained by (14). Their corresponding external forces are obtained by (12) and (13). The desired external forces are not always equal to the GRFs because the bipedal robot is an unactuated system and the supporting convex hull (SCH) is limited (see Fig. 2). If the desired ZMP exceeds the SCH, the ankles would have an unexpected rotation. Friction is related to the vertical force and friction coefficient; so the desired friction, divided by the desired vertical force, must be smaller than the friction coefficient. If it is not, the feet of the robot slip. The ground can only provide the unilateral force vertically, so the vertical force must be positive. The ground can only push the feet. All in all, actual external forces satisfy these constraints. Therefore, as long as the desired external forces satisfy these constraints, we can choose extremal accelerations, which enable the robot to recover from disturbances as best as possible.

5.1. Constraints on accelerations of the floating base

First, the relationship between the accelerations of the floating base and their corresponding external forces is deduced. Rewrite $M(q)$ as follows:

$$M(q) = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix}.$$

Each element of this matrix is an $R^{6\times6}$ matrix. The following equation is obtained by (7):

$$M_{11}\ddot{q}_r + M_{12}\ddot{q}_{\mathrm{rl}} + M_{13}\ddot{q}_{\mathrm{ll}} + C_1 = {}^{W}J_{M1}^{T}{}^{M}X_{W}^{T}{}^{M}\hat{f}_m, \tag{16}$$

https://doi.org/10.1017/S0263574713000829 Published online by Cambridge University Press

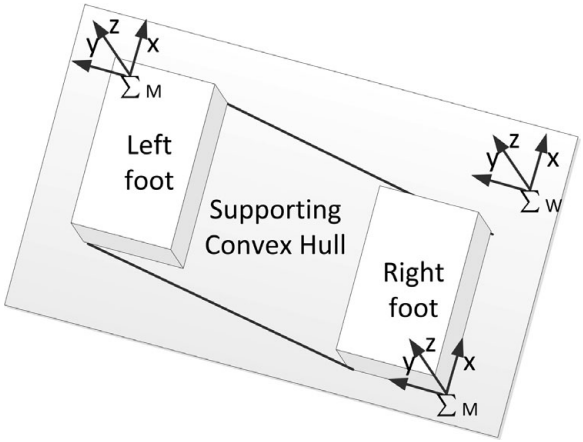


Fig. 2. Supporting convex hull.

where ${}^{M}X_{W}$ is the coordinate transformation matrix from $\sum_W$ to frame $\sum_M$ (see Fig. 2). $\sum_M$ is selected to locate where GRFs provide the maximum or minimum accelerations. ${}^{M}\hat{f}_m$ is the desired external force expressed in this frame.

Equation (16) is simplified as follows by (9) and (10):

$$H\ddot{q}_r + D = W {}^{M}\hat{f}_m, \tag{17}$$

where

$$W = {}^{W}J_{M1}^{T} {}^{M}X_{W}^{T}, \tag{18}$$

$$H = M_{11} - M_{12} {}^{W}J_{\mathrm{rf2}}^{-1\,W}J_{\mathrm{rf1}} - M_{13} {}^{W}J_{\mathrm{lf3}}^{-1\,W}J_{\mathrm{lf1}}, \tag{19}$$

$$D = C_1 - M_{12} {}^{W}J_{\mathrm{rf2}}^{-1\,W}\dot{J}_{\mathrm{rf}}\dot{q} - M_{13} {}^{W}J_{\mathrm{lf3}}^{-1\,W}\dot{J}_{\mathrm{lf}}\dot{q}. \tag{20}$$

The following equation is obtained by (8), (11) and (17):

$$Ua^{c} + V = {}^{M}\hat{f}_m, \tag{21}$$

where

$$U = W^{-1}H {}^{W}J_{R}^{-1\,R}X_{W}^{-1}, \tag{22}$$

$$V = W^{-1}\left(D - H {}^{W}J_{R}^{-1}\left({}^{R}X_{W}^{-1}[0_{3\times1}; {}^{R}\omega_r \times {}^{R}v_r] + {}^{W}\dot{J}_R\dot{q}_r\right)\right). \tag{23}$$

Equation (21) shows a clear relationship between the desired conventional acceleration and the desired external force. U and V are functions of q and $\dot{q}$, which means that different poses and velocities cause different U and V. Thus, U and V are updated in each control period.

Note that ${}^{M}\hat{f}_m$ is a spatial vector which has six elements, ${}^{M}\hat{f}_m = [n_{Mx}, n_{My}, n_{Mz}, f_x, f_y, f_z]^T$. The force constraints, mentioned above, can be further understood through the following inequalities:

- $f_z > 0$. The ground can only push the feet.
- $\frac{f_x}{f_z} < \frac{\mu}{\sqrt{2}}$ and $\frac{f_y}{f_z} < \frac{\mu}{\sqrt{2}}$. No slipping happens. μ is the coefficient of friction.
- $n_{My} \geq or \leq 0$ guarantees the desired ZMP within the SCH in the x direction. Choosing $\geq$ or $\leq$ depends on where the coordinate frame is. For instance, if the frame is at the front side of the SCH, $\geq$ is selected; if the frame is at the back side of the SCH, $\leq$ is selected.
- $n_{Mx} \geq or \leq 0$ guarantees the desired ZMP within the SCH in the y direction. If the frame is at the left side of the SCH, $\leq$ is selected; if the frame is at the right side of the SCH, $\geq$ is selected.

https://doi.org/10.1017/S0263574713000829 Published online by Cambridge University Press

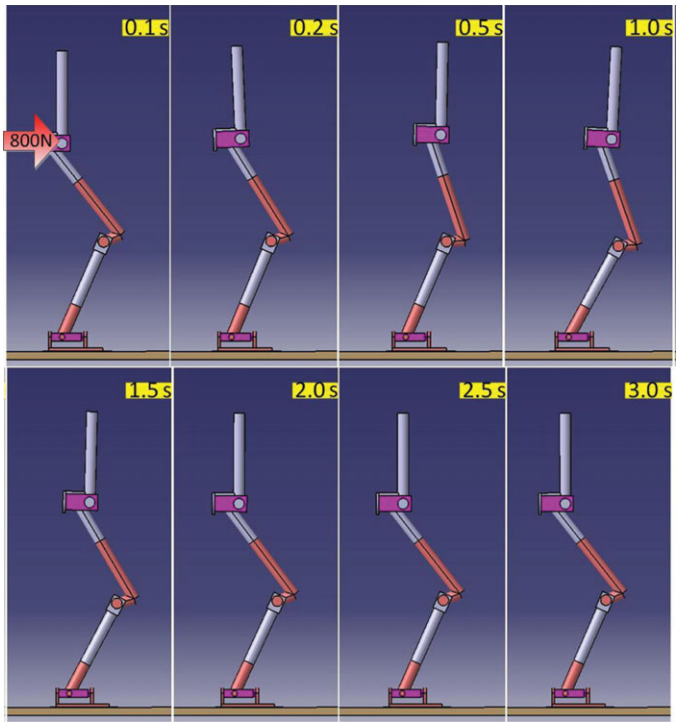


Fig. 3. (Colour online) Screen shot of the robot recovering from a push with 800 N for 0.1 s.

There are two constrained ${}^{M}\hat{f}_m$ which are expressed in two $\sum_M$ (see Fig. 2), respectively. According to the force constraints mentioned above and (21), we can obtain five inequalities for each ${}^{M}\hat{f}_m$. These 10 inequalities are related to U, V and ${}^{M}\hat{f}_m$. We rewrite them into matrix form as

$$A * a^c_{act} \leq b, \tag{24}$$

where $A \in R^{10\times 3}$ and $b \in R^{10\times 1}$ are functions of U, V and ${}^{M}\hat{f}_m$, respectively, a^c_{act} is the actual acceleration which should satisfy its constraints.

5.2. Quadratic programming

When we get desired acceleration a^c_{des} by (14), QP is used to find extremal ones. The cost function is

$$f(a^c_{\text{act}}) = \left(a^c_{\text{act}} - a^c_{\text{des}}\right)^T * W_t * \left(a^c_{\text{act}} - a^c_{\text{des}}\right), \tag{25}$$

where W_t is a weight matrix.

Finally, $f(a^c_{\text{act}})$ is minimized as

$$\min_{a^c_{\text{act}}} f\left(a^c_{\text{act}}\right),\ s.t.\ A * a^c_{\text{act}} \leq b \tag{26}$$

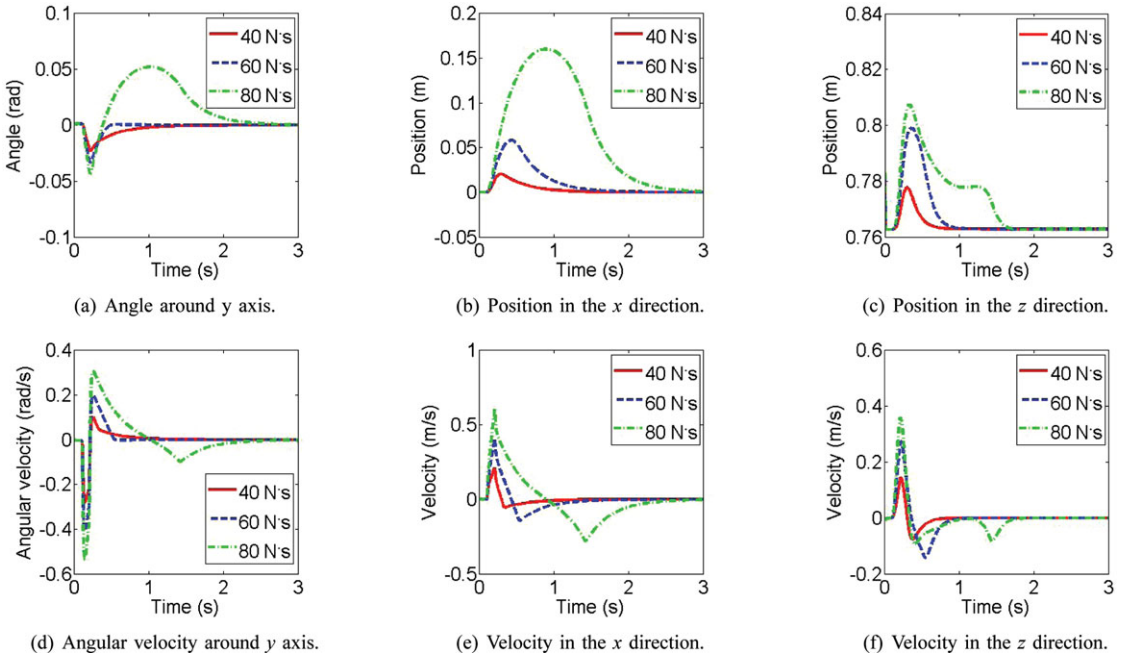
to obtain extremal accelerations, which are used instead of a^c_{des} to perform inverse dynamics.

6. Results of Experiments

Like the methods shown in Table I, different push forces in the x direction are selected to evaluate our method. Compared with pushes in other directions, pushes along the x axis more easily cause the robot to tip over. This method is suitable to impulsive push, so all pushes in the experiments last for 0.1 s.

The robot stands in place and keeps its torso upright. It has the same mass distribution as the real robot. The total mass of the robot is 96.9 kg. The torso has more than 70% of the total mass. So the

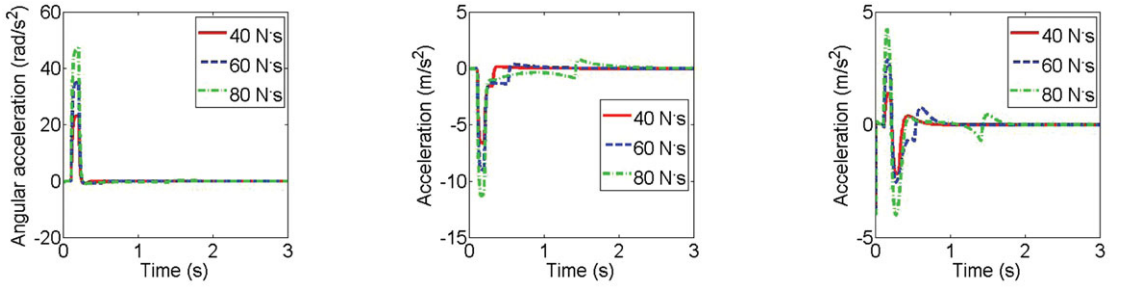
https://doi.org/10.1017/S0263574713000829 Published online by Cambridge University Press



(a) Angle around y axis. (b) Position in the x direction. (c) Position in the z direction.

(d) Angular velocity around y axis. (e) Velocity in the x direction. (f) Velocity in the z direction.

Fig. 4. (Colour online) The motion of the robot's torso when it suffers from different pushes.



(a) Angular acceleration around y axis. (b) Acceleration in the x direction. (c) Acceleration in the z direction.

Fig. 5. (Colour online) Actual accelerations when the robot suffers from different pushes.

torso represents the CoM approximately. The length of the forefoot is 0.182 m. The height of the CoM is 0.883 m. It is lower than in ref. [5] because our model squats down about 0.07 m initially.

Figure 3 shows the screen shots of the robot recovering from a disturbance. At moment 0.1 s, apply a force with 800 N at the pelvis. Then, the robot deviates from its original position and velocity. Our method enables the robot to move its whole body to recover to its original position and attitude.

In order to study the influence of different forces on torso motion, we do a group of simulations with 400, 600 and 800 N applied at the pelvis. The motion in the lateral plane influences the motion in the sagittal plane, but not much, so Figs. 4 and 5 only show the motion in the sagittal plane. If the push is small, the robot recovers quickly. If the push is large, the robot recovers slowly. Additionally, the push in the x direction also causes torso motion in the z direction and around the y axis. It means that our method uses whole-body motion for push recovery. Figure 6 shows the actual and desired accelerations when the robot suffers from 800 N. As mentioned in Section 4, the constraints on the accelerations of the floating base are related to joint angles and angular velocities, so the extremal accelerations vary over the whole-body motion. In the beginning, the desired accelerations are beyond the acceleration constraints. We choose the accelerations on the boundary which are the extremal ones. After a while, the desired accelerations are under the acceleration constraints. Then, the optimized accelerations are equal to the desired ones. Figure 7 shows the motion of the torso when selecting different weight matrix W_t. A bigger coefficient in an axis means more accurate tracking in this axis but less contribution to push recovery. We choose three different W_t which affects the

https://doi.org/10.1017/S0263574713000829 Published online by Cambridge University Press

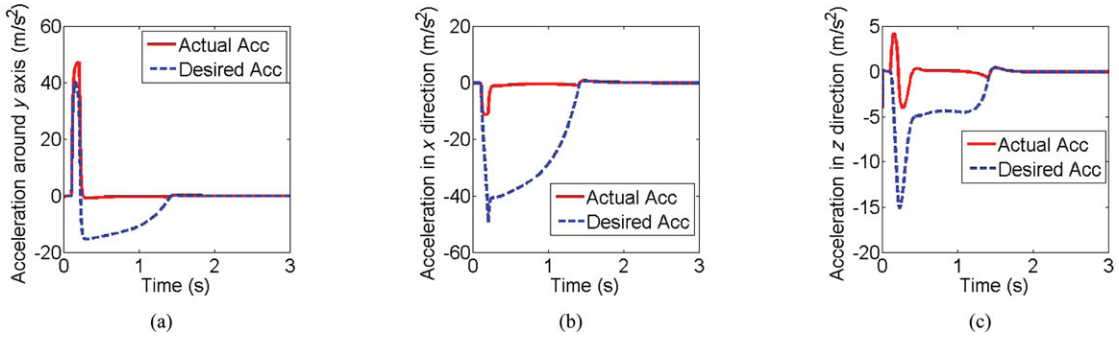


Fig. 6. (Colour online) Actual and desired accelerations when the robot suffers from 800 N.

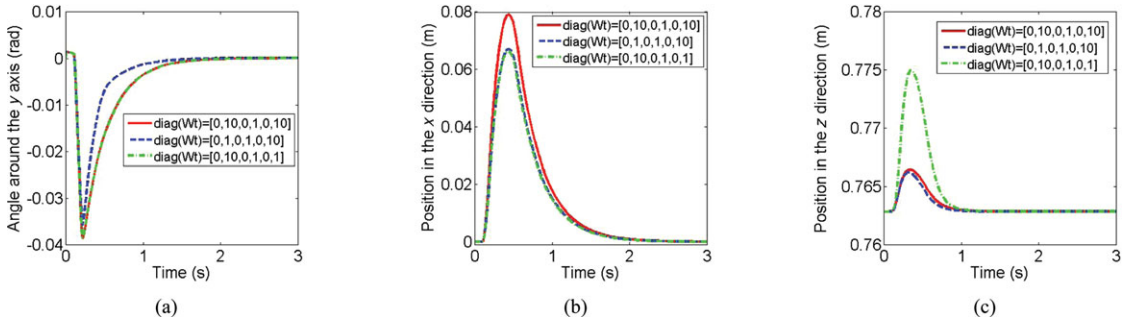


Fig. 7. (Colour online) The motion of the robot's torso when it suffers from 600 N with different weight matrix W_t.

motion in the sagittal plane. Note that diag(W_t) = [0, 10, 0, 1, 0, 10] shows that the motion in the z direction and around the y axis has little contribution to push recovery. diag(W_t) = [0, 1, 0, 1, 0, 10] shows that the motion around the y axis has much contribution. diag(W_t) = [0, 10, 0, 1, 0, 1] shows that the motion in the z direction has much contribution. We know from Fig. 7(b) that the deviation in the x direction is bigger if the motion in/around the other axes has less contribution to push recovery. It is clear from Figs. 7(a) and (b) that the motion around the y axis contributes to push recovery by comparing the red solid line and the blue dashed line. Figs. 7(b) and (c) show that the motion in the z direction contributes to push recovery by comparing the red solid line and the green dash–dotted line.

We increase the push force when we do simulations. When the push force is over 800 N, it is hard for the robot to recover to its original position even through we vary the weight matrix W_t. So we use a conservative value 800 N to calculate the robustness of our method. It is 1.36. Our model is similar to a three-link model, because it does not have arms. The robustness of our method would be higher if our method is applied on a four-link model.

7. Conclusion

This paper has presented a robust push recovery method by whole-body dynamics control. The relationship between the accelerations of the floating base and the desired external forces was revealed. Using this relationship and selecting accelerations on the boundary of the acceleration inequalities, our method enables the robot to move its whole body to recover from pushes as best as possible. According to the proposed robustness criterion, our method is more robust than presented methods. The effectiveness of the proposed method is demonstrated by push recovery on a force-controlled bipedal robot in simulation.

Acknowledgment

This work was supported by the National High Technology Research of China under grant 2011AA040202, the National Natural Science Foundation of China under grants 60925014, 61273348

https://doi.org/10.1017/S0263574713000829 Published online by Cambridge University Press

and 61175077, Beijing Science Foundation under grant 4122065, and the "111 Project" under grant B08043.

References

1. J. Pratt and B. Krupp, "Design of a bipedal walking robot," *Proc. SPIE* **6962**, 69621F (1–13) (2008).
2. B. Stephens and C. Atkeson, "Dynamic Balance Force Control for Compliant Humanoid Robots," **In**: *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan (2010) pp. 1248–1255.
3. S.-H. Hyon and G. Cheng, "Passivity-Based Full-Body Force Control for Humanoids and Application to Dynamic Balancing and Locomotion," **In**: *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China (2006) pp. 4915–4922.
4. Boston Dynamics, "PETMAN," available at: http://www.bostondyna-mics.com/robot_petman.html (Oct. 2009)
5. B. J. Stephens and C. G. Atkeson, "Push Recovery by Stepping for Humanoid Robots with Force Controlled Joints," **In**: *Proceedings of IEEE International Conference on Humanoid Robots*, Nashville, TN (2010), pp. 52–59.
6. H. Diedam, D. Dimitrov, P.-B. Wieber, K. Mombaur and M. Diehl, "Online Walking Gait Generation with Adaptive Foot Positioning Through Linear Model Predictive Control," **In**: *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, Nice, France (2008) pp. 1121–1126.
7. P.-B. Wieber, "Trajectory Free Linear Model Predictive Control for Stable Walking in the Presence of Strong Perturbations," **In**: *Proceedings of the IEEE International Conference on Humanoid Robots*, Genoa, Italy (2006) pp. 137–142.
8. A. Kuo, "An optimal control model for analyzing human postural balance," *IEEE Trans. Biomed. Eng.* **42**, 87–101 (1995).
9. C. Atkeson and B. Stephens, "Multiple Balance Strategies from One Optimization Criterion," **In**: *Proceedings of the IEEE International Conference on Humanoid Robots*, Pittsburgh, PA (2007) pp. 57–64.
10. X. Dengpeng and L. Xu, "Multiple balance strategies for humanoid standing control," *Acta Autom. Sin.* **37**(2), 228–233 (2001).
11. C. G. Liu and C. G. Atkeson, "Standing Balance Control Using a Trajectory Library," **In**: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, MO (2009) pp. 3031–3036.
12. B. Stephens, "Humanoid Push Recovery," **In**: *Proceedings of the IEEE International Conference on Humanoid Robots* (2007) pp. 589–595.
13. T. Qing, X. Rong and C. Jian, "Tip over avoidance control for biped robot," *Robotica* **27**, 883–889 (2009).
14. H. Han, T. Kim and T. Park, "Tolerance Analysis of a Spur Gear Train," **In**: *Proceedings of the 3rd DADS Korean User's Conference*, Seoul, Korea (1987) pp. 61–81.
15. F. Roy, "A beginner's guide to 6-D vectors (Part 1)," *IEEE Robot. Autom. Mag.* **17**(3), 83–94 (2010).
16. F. Roy, "A beginner's guide to 6-D vectors (Part 2)," *IEEE Robot. Autom. Mag.* **17**(4), 88–99 (2010).