# Stability Control of Human Inspired Jumping Robot

**Jan Babič, Damir Omrčen**

Department of Automatics, Biocybernetics and Robotics
"Jožef Stefan" Institute
Ljubljana, Slovenia
E-mail: jan.babic@ijs.si

***Abstract:*** *The purpose of this study is to describe the necessary conditions for the motion controller of a humanoid robot to perform the vertical jump. We performed vertical jump simulations using three different control algorithms and showed the effects of each algorithm on the vertical jump performance. We showed that motion controllers which consider one of two conditions separately are not appropriate to control the vertical jump. We demonstrated that the motion controller has to satisfy both conditions simultaneously in order to achieve a desired vertical jump.*

***Keywords:*** *humanoid robot, vertical jump, dynamic stability*

## I INTRODUCTION

The vertical jump is an example of a fast explosive movement that requires quick and completely harmonized coordination of all segments of the robot, for the push-off, for the flight and, finally, for the landing. The most important part of the vertical jump which influences the efficiency and therefore the height of the jump is the push-off phase. The push-off phase can be defined as a time interval when the feet are touching the ground before the flight. The primary task of the actuators during the push-off phase is to keep the robot balanced during the entire jump. The secondary task of the actuators is to accelerate the robot's center of mass upwards in the vertical direction to the extended body position.

In the past, several research groups developed and studied jumping robots but most of these were simple mechanisms not similar to humans. They were controlled by empirically derived control strategies. Probably the best-known hopping robots were designed by Raibert and his team [2]. They developed different hopping robots, all with telescopic legs and with a steady-state control algorithm. Later, De Man et al. developed a trajectory generation strategy based on the angular momentum theorem which was implemented on a model with articulated legs [1]. Recently Hyon et al. developed a one-legged hopping robot with a structure based on the hind-limb model of a dog [4]. They used an empirically derived controller based on the characteristic dynamics.

The purpose of this study is to mathematically formulate the necessary conditions that the motion controller of a humanoid robot has to consider in order to perform the vertical jump.

## II DYNAMICAL MODEL OF JUMPING ROBOT

The model of the jumping robot is planar and is composed of four segments which represent the foot, shank, thigh and trunk. The segments are connected by frictionless rotational hinges whose axes are perpendicular to the sagital plane. The model consists of two parts, the model of the robot in the air and the model of the robot in contact with the ground. While the tip of the foot is on the ground, the contact between the foot tip and the ground is modeled as a rotational hinge joint between the foot tip and the ground at point F. Therefore, the robot has six degrees of freedom during flight and four degrees of freedom during stance (with the assumption that the foot tip of the robot does not slip and does not bounce back). The generalized coordinates used to describe the motion of the robot are coordinates $x_F$ and $y_F$ of the foot tip measured in the reference frame and joint angles $\alpha$, $\beta$, $\gamma$, $\delta$.

## III VERTICAL JUMP CONDITIONS AND CONTROL ALGORITHM

To assure the verticality of the jump, the robot's center of mass (COM) has to move in the upward direction above the support polygon during the push-off phase of the jump. The second condition, which refers to the balance of the robot during the push-off phase, is the position of the zero moment point (ZMP). ZMP is the point on the ground at which the net moment of the inertial forces and the gravity forces has no component along the horizontal axes [3]. In the following sections we will analyze how these two conditions influence the vertical jump. First we will design two control algorithms based on the COM condition and ZMP condition separately and then we will design a control algorithm that considers both conditions together.

Equations that define the position of COM are

$$x_{com} = \frac{\sum_{i=1}^{n} m_i x_i}{\sum_{i=1}^{n} m_i}, y_{com} = \frac{\sum_{i=1}^{n} m_i y_i}{\sum_{i=1}^{n} m_i}, \tag{1}$$

where $x_{com}$ and $y_{com}$ are horizontal and vertical positions of COM of the whole system, respectively. $x_i$ and $y_i$ are the coordinates of COM of the $i$-th segment, $m_i$ is the mass of the $i$-th segment and $n$ is the number of segments.

The position of ZMP is

$$x_{zmp} = \frac{\sum_{i=1}^{n} m_i x_i (\ddot{y}_i + g) - \sum_{i=1}^{n} m_i y_i \ddot{x}_i + \tau_z}{\sum_{i=1}^{n} m_i (\ddot{y}_i + g)}, \tag{2}$$

where

$$\tau_z = \sum_{i=1}^{n} (\mathbf{I}_i \dot{\omega}_i + \omega_i \times \mathbf{I}_i \omega_i). \tag{3}$$

$g$ is the quadratic norm of the gravity vector, $\mathbf{I}_i$ is the inertial tensor of the $i$-th segment around its COM and $\omega_i$ is the angular velocity of the $i$-th segment. When the robot is at rest, the position of ZMP coincides with the horizontal position of COM.

For the control purposes we have to find the second derivatives of $x_{com}$ and $y_{com}$ (Eq. 1). We get the following equations

$$\ddot{x}_{com} = k_{11}\ddot{\alpha} + k_{12}\ddot{\beta} + k_{13}\ddot{\gamma} + k_{14}\ddot{\delta} + d_1 \tag{4}$$

and

$$\ddot{y}_{com} = k_{21}\ddot{\alpha} + k_{22}\ddot{\beta} + k_{23}\ddot{\gamma} + k_{24}\ddot{\delta} + d_2, \tag{5}$$

where the parameters $k_{ij}$ and $d_i$ are functions of joint angles ($k_{ij} = f(\alpha,\beta,\gamma,\delta), d_i = f(\alpha,\beta,\gamma,\delta)$).

The position of ZMP on the ground can not be described in this form because the denominator of Eq. 2 is also a function of joint angles. However, in many cases we can freely move the coordinate system to coincide with the position of the desired ZMP and the balancing condition becomes $x_{zmp} = 0$. In this case we can express $x_{zmp}$ as

$$x_{zmp} = 0 = k_{31}\ddot{\alpha} + k_{32}\ddot{\beta} + k_{33}\ddot{\gamma} + k_{34}\ddot{\delta} + d_3. \qquad (6)$$

Eqs. 4, 5 and 6 can be combined and written in the matrix form

$$\begin{bmatrix} \ddot{x}_{com} \\ \ddot{y}_{com} \\ 0 \end{bmatrix} = \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} \\ k_{21} & k_{22} & k_{23} & k_{24} \\ k_{31} & k_{32} & k_{33} & k_{34} \end{bmatrix} \begin{bmatrix} \ddot{\alpha} \\ \ddot{\beta} \\ \ddot{\gamma} \\ \ddot{\delta} \end{bmatrix} + \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix}, \qquad (7)$$

where $\ddot{x}_{com}$ and $x_{zmp}$ are the conditions that relate with the balance. On the other hand, $\ddot{y}_{com}$ is the prescribed vertical acceleration of the robot's COM during the push-off phase of the jump which enables the robot to jump.

**Control of $x_{com}$**

In the first case we analyse the vertical jump when the motion controller keeps the horizontal position of the robot's COM over the virtual joint connecting the foot with the ground at point F during the entire push-off phase of the vertical jump. Motion controller does not control the position of ZMP $x_{zmp}$. By rewriting Eq. 7 for $x_{com}$ and $y_{com}$ we get

$$\begin{bmatrix} \ddot{x}_{com} \\ \ddot{y}_{com} \end{bmatrix} = \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} \\ k_{21} & k_{22} & k_{23} & k_{24} \end{bmatrix} \begin{bmatrix} \ddot{\alpha} \\ \ddot{\beta} \\ \ddot{\gamma} \\ \ddot{\delta} \end{bmatrix} + \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}. \qquad (8)$$

Since the system is under-determinate (the degree of redundancy is two), we have to set up two additional constraints. To achieve a human like motion of the vertical jump we chose the following simple constraints

$$\ddot{\gamma} = c_1\ddot{\beta}, \ddot{\delta} = c_2\ddot{\beta}, \qquad (9)$$

where $c_1$ and $c_2$ are constants. By substitution of Eq. 9 into Eq. 8 we get

$$\begin{bmatrix} \ddot{x}_{com} \\ \ddot{y}_{com} \end{bmatrix} = \begin{bmatrix} k_{11} & k_{12}+c_1k_{13}+c_2k_{14} \\ k_{21} & k_{22}+c_1k_{23}+c_2k_{24} \end{bmatrix} \begin{bmatrix} \ddot{\alpha} \\ \ddot{\beta} \end{bmatrix} + \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}. \qquad (10)$$

The system of equations is determinate and the joint accelerations can be written as

$$\begin{bmatrix} \ddot{\alpha} \\ \ddot{\beta} \end{bmatrix} = \begin{bmatrix} k_{11} & k_{12}+c_1k_{13}+c_2k_{14} \\ k_{21} & k_{22}+c_1k_{23}+c_2k_{24} \end{bmatrix}^{-1} \left( \begin{bmatrix} \ddot{x}_{com} \\ \ddot{y}_{com} \end{bmatrix} - \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \right). \qquad (11)$$

**Control of $x_{zmp}$**

In the second case we analyse the vertical jump when the motion controller keeps the position of ZMP aligned with the virtual joint at point F. The motion controller does not control the horizontal position of COM ($x_{com}$). By rewriting Eq. 7 for $x_{zmp}$ and $y_{com}$ we get

$$\begin{bmatrix} \ddot{y}_{com} \\ 0 \end{bmatrix} = \begin{bmatrix} k_{21} & k_{22} & k_{23} & k_{24} \\ k_{31} & k_{32} & k_{33} & k_{34} \end{bmatrix} \begin{bmatrix} \ddot{\alpha} \\ \ddot{\beta} \\ \ddot{\gamma} \\ \ddot{\delta} \end{bmatrix} + \begin{bmatrix} d_2 \\ d_3 \end{bmatrix}. \qquad (12)$$

Similarly as in the previous case we have to find the joint accelerations. If we again use the same constraints (9) we get the following determinate system of equations

$$\begin{bmatrix} \ddot{y}_{com} \\ 0 \end{bmatrix} = \begin{bmatrix} k_{21} & k_{22}+c_1k_{23}+c_2k_{24} \\ k_{31} & k_{32}+c_1k_{33}+c_2k_{34} \end{bmatrix} \begin{bmatrix} \ddot{\alpha} \\ \ddot{\beta} \end{bmatrix} + \begin{bmatrix} d_2 \\ d_3 \end{bmatrix}, \qquad (13)$$

and the joint accelerations are

$$\begin{bmatrix} \ddot{\alpha} \\ \ddot{\beta} \end{bmatrix} = \begin{bmatrix} k_{21} & k_{22}+c_1k_{23}+c_2k_{24} \\ k_{31} & k_{32}+c_1k_{33}+c_2k_{34} \end{bmatrix}^{-1} \left( \begin{bmatrix} \ddot{x}_{com} \\ \ddot{y}_{com} \end{bmatrix} - \begin{bmatrix} d_2 \\ d_3 \end{bmatrix} \right). \qquad (14)$$

**Control of $x_{com}$ and $x_{zmp}$**

In the third case we will analyse the vertical jump when the motion controller considers both conditions from the precedent two sections. It keeps the position of ZMP and the horizontal position of the robot's COM aligned with the virtual joint at point F. In this case the degree of redundancy is one. The following constraint that abolishes the redundancy of Eq. 7 is the relationship of the ankle and knee joint accelerations

$$\ddot{\gamma} = C_1\ddot{\beta} \tag{15}$$

where $C_1$ is a constant. By substitution of Eq. 15 into Eq. 7 we get

$$\begin{bmatrix} \ddot{x}_{com} \\ \ddot{y}_{com} \\ 0 \end{bmatrix} = \begin{bmatrix} k_{11} & k_{12}+C_1k_{13} & k_{14} \\ k_{21} & k_{22}+C_1k_{23} & k_{24} \\ k_{31} & k_{32}+C_1k_{33} & k_{34} \end{bmatrix} \begin{bmatrix} \ddot{\alpha} \\ \ddot{\beta} \\ \ddot{\delta} \end{bmatrix} + \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix}, \tag{16}$$

and the joint accelerations are

$$\begin{bmatrix} \ddot{\alpha} \\ \ddot{\beta} \\ \ddot{\delta} \end{bmatrix} = \begin{bmatrix} k_{11} & k_{12}+C_1k_{13} & k_{14} \\ k_{21} & k_{22}+C_1k_{23} & k_{24} \\ k_{31} & k_{32}+C_1k_{33} & k_{34} \end{bmatrix}^{-1} \left( \begin{bmatrix} \ddot{x}_{com} \\ \ddot{y}_{com} \\ 0 \end{bmatrix} - \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} \right). \tag{17}$$

### Motion Controller

For the control of the robot we used a simple feed forward joint acceleration controller

$$\tau_c = \mathbf{H}(q)\ddot{q}_c + \mathbf{C}(\dot{q},q) + g(q), \tag{18}$$

where $\tau_c$ and $q$ denote the control torque and the vector of joint positions, respectively. $\mathbf{H}$, $\mathbf{C}$ and $g$ denote the inertia matrix, the vector of Coriolis and centrifugal forces and the vector of gravity forces, respectively. $\ddot{q}_c$ is the vector of control accelerations ($\ddot{q}_c = \left[\ddot{\alpha}, \ddot{\beta}, \ddot{\gamma}, \ddot{\delta}\right]^T$). During the push-off phase of the jump $\ddot{q}_c$ is defined by Eqs. (11), (14) or (17). During the flight phase, when the robot is in the air, the angular momentum and the linear momentum are conserved and the $\ddot{q}_c$ is set in such a way that the joint motions stops and the robot is prepared for landing.

## IV SIMULATION STUDY

We performed vertical jump simulations using three different control algorithms described in the previous section. First we simulated the vertical jump using the control algorithm based on the COM condition, then we simulated the vertical jump using the control algorithm based on the ZMP condition and, finally, we simulated the jump where the controller considered both conditions together.

### Control of $x_{com}$

In this case we controlled $\ddot{y}_{com}$ and $\ddot{x}_{com}$ as defined by Eq. 11. From the requirement that $\ddot{x}_{com}$ has to be above the support polygon (point F) follows that $x_{com} = 0$ and $\ddot{x}_{com} = 0$. Fig. 1 shows the position of COM during the jump. The solid line represents the horizontal position while the dashed line represents the vertical position of COM. Dotted line shows the moment of take-off. It is evident that the horizontal position of COM remains zero, i.e. COM is above point F.

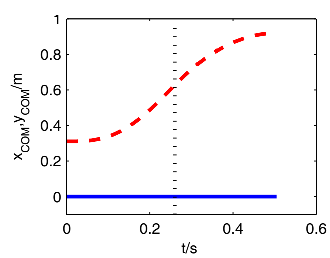


Figure 1
Position of center of mass during vertical jump considering only COM condition

Due to the fact that we did not control the position of ZMP, the required torque in the virtual joint between the foot and the ground during the push-off phase of the jump is not zero (see Fig. 2). As this torque can not be applied to the real robotic system, this controller is not appropriate for performing the vertical jump. Without applying this torque at the virtual joint the robot becomes unbalanced.

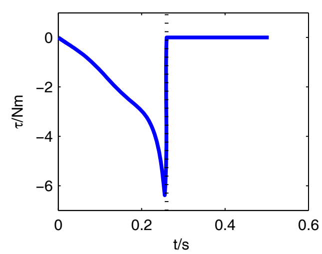


Figure 2
Required torque in virtual joint considering only COM condition

## Control of $x_{zmp}$

In this case we controlled $\ddot{y}_{com}$ and $x_{zmp}$, as defined by Eq. 14. To satisfy the balance criteria $x_{zmp}$ has to be over the support polygon ($x_{zmp} = 0$). As evident from Fig. 3, the horizontal position of COM during the push-off phase of the jump is not zero and, therefore, the robot does not perform the vertical jump as it should.

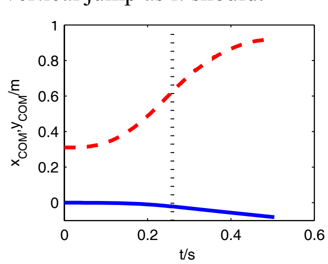


Figure 3
Position of center of mass during vertical jump considering only ZMP condition

On the other hand, the torque in the virtual joint is zero (Fig. 4) and the system is balanced without the torque in the virtual joint between the foot and the ground. Therefore, the robot performs a jump, but this is not a vertical jump, since COM is not above point F at the take-off moment.

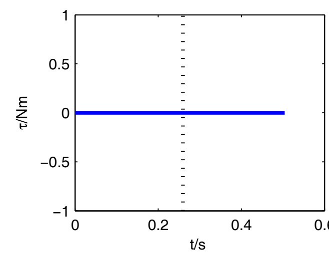


Figure 4
Required torque in virtual joint considering only ZMP condition

## Control of $x_{com}$ and $x_{zmp}$

In this case we controlled $\ddot{y}_{com}$ together with both $\ddot{x}_{com}$ and $x_{zmp}$, as defined by Eq. 17. Fig. 5 shows the position of COM during the jump and Fig. 6 shows the torque in the virtual joint.

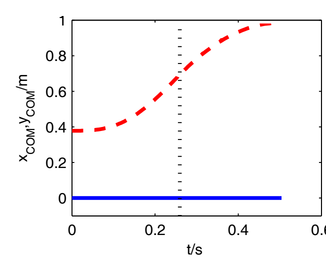


Figure 5
Position of center of mass during vertical jump considering both COM and ZMP conditions

As the position of COM is always above point F and the torque in the virtual joint is zero, the robot performs the desired vertical jump. Therefore, both conditions have to be fulfilled to assure the verticality of the jump. Both, the horizontal position of COM and the position of ZMP have to coincide with point F.

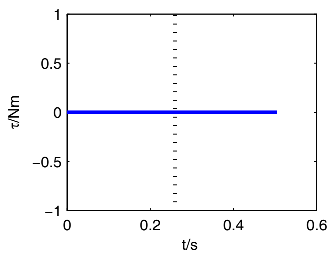


Figure 6
Torque in virtual joint considering both COM and ZMP

## Conclusions

In this study, we mathematically formulated the necessary conditions which have to be considered by the motion controller to perform the vertical jump. The first condition refers to the robot's center of gravity which has to move in the upward direction above the support polygon during the push-off phase of the jump. The second condition refers to the position of the zero moment point that has to lie inside the support polygon to assure the balance of the robot. We analyzed how these two conditions influence the vertical jump performance. Based on these conditions we designed three different control algorithms and used them in vertical jump simulations. We showed that motion controllers that consider one of two conditions separately are not appropriate for the control of the vertical jump. We demonstrated that the motion controller has to satisfy both conditions simultaneously in order to achieve a desired vertical jump.


## Acknowledgement

This study was supported by the Slovenian Ministry of Higher Education, Science and Technology.


## References


[1] De Man H., Lefeber D., Daerden F., Faignet E.: Simulation of a new control algorithm for a one-legged hopping robot (using the multibody code mechanica motion). In Proceedings International Workshop on Advanced Robotics and Intelligent Machines, pp. 1-13, Manchester, UK, 1996

[2] Raibert M.: Legged Robots That Balance. MIT Press, 1986

[3] Vukobratović M., Borovac B.: Zero-moment point - thirty five years of its life. International Journal of Humanoid Robotics, 1(1):157-173, 2004

[4] Hyon S., Emura T., Mita T.: Dynamics-based control of a one-legged hopping robot. Journal of Systems and Control Engineering, 217(2):83-98, 2003