Enhancing ZMP-based Control Model: System Identification Approach

Wael Suleiman, Fumio Kanehiro, Kanako Miura and Eiichi Yoshida

CNRS-AIST JRL (Joint Robotics Laboratory), UMI3218/CRT
Intelligent Systems Research Institute
National Institute of Advanced Industrial Science and Technology (AIST)
Tsukuba Central 2, 1-1-1 Umezono, Tsukuba, Ibaraki, 305-8568 Japan
{wael.suleiman, f-kanehiro, kanako.miura, e.yoshida}@aist.go.jp

Abstract

The approximation of humanoid robot by an inverted pendulum is one of the most used models to generate a stable walking patterns using a planned Zero Moment Point (ZMP) trajectory. However, on account of the difference between the multi-body model of the humanoid robot and the simple inverted pendulum model, the ZMP error might be bigger than the polygon of support and the robot falls down. To overcome this limitation, we propose to improve the accuracy of inverted pendulum model using system identification techniques. The candidate model is a quadratic in the state space representation. To identify this system, we propose an identification method which is the result of the comprehensive application of system identification to dynamic systems. Based on the quadratic system, we propose as well a controlling algorithms for on-line and off-line walking patterns generation for humanoid robots. The efficiency of the quadratic system and the walking patterns generation methods has been successfully shown using dynamical simulation and conducting real experiments on the cybernetic human HRP-4C.

keywords: Humanoid robot; ZMP control; System identification; Optimization; Nonlinear system control

1 Introduction

Research on biped humanoid robot is nowadays one of the most active research fields in robotics. Making the humanoid robots walk was the subject of intensive investigation last years. Many researchers have proposed different methods to generate a stable walking motion for humanoid robot [1–8]. Most of these
methods are using a simplified model which it is based on the approximation of the humanoid robot by an inverted pendulum model. The mass of inverted pendulum coincide with the Centre of Mass (CoM) of the humanoid robot.

In order to generate a stable motion (dynamically balanced), those methods use the principle of Zero Moment Point (ZMP) [9]. The ZMP is a point of the support polygon (i.e. the convex hull of all points of contact between the support foot (feet) and the ground), at which the horizontal moments are vanishing.

The cart-table model proposed by Kajita et al [3] is one of these methods. The efficiency of this method has been proved by generating on-line stable walking motions using a planned trajectory of ZMP.

Some methods have been proposed to compensate the error between the simple inverted pendulum model and the multi-body model of the humanoid robot [10,11]. To achieve the same goal, the purpose in this paper is however to show that the linear model related to cart-table model can be improved by capturing the dynamic difference between the the multi-body model of the humanoid robot and that of the cart-table model, then identifying a quadratic system [12] which is able to approximate this difference. Furthermore, we aim at giving insights into the problem of controller design of quadratic system in order to generate dynamically stable walking patterns. The dynamic balance of the humanoid robot during the execution of the ZMP trajectory is subject to several perturbations such as inaccuracy of the dynamic and kinematics parameters and the flexibility in some joints of the humanoid robot. However, by guaranteeing that the ZMP trajectory is always inside a safety margin of stability, the online stabilizer can ensure the dynamic balance of the humanoid robot. Therefore, the dynamical stability is guaranteed by tracking a pre-defined ZMP trajectory which is always inside of the polygon of support while respecting a safety margin.

The proposed controlling algorithms are classified into two categories:

1. Off-line walking pattern generation algorithms: in this category the computational time is not considered, and the objective is to track the desired ZMP trajectory precisely.

2. On-line walking pattern generation algorithms: in this category the computational complexity of the controller should respect the real-time constraints. In this case, the desired ZMP trajectory is tracked as much as possible.

The paper is organized as follows. An overview of the cart-table model and the proposed model to improve the reliability of the cart-table model is discussed in Section 2. An adapted identification method for the identification of quadratic system is developed in Section 3. Section 4 points out the problem of designing a controller to track a desired ZMP trajectory. The efficiency of the proposed
quadratic system and the walking generation algorithms is shown in Section 5 through real experiment on the cybernetic human HRP-4C [13].

2 Cart-table Model and Quadratic System

The main idea of cart-table model [3] is to approximate the humanoid robot by a mass located at its Center of Mass (CoM) and it is equal to the total mass of the humanoid robot. Therefore, the complex problem of controlling the humanoid robot is transformed to control an inverted pendulum.

Let us define the Cartesian position of the center of mass ($P_{CoM}$) by

$$P_{CoM} = \begin{bmatrix} x \\ y \\ z_c \end{bmatrix} \quad (1)$$

We suppose that the vertical position of the mass $z_c$ is constant.

The ZMP is a particular point of the horizontal plane at which the horizontal moments vanish. For the inverted pendulum, it is defined as follows

$$ZMP = \begin{bmatrix} p_x \\ p_y \end{bmatrix} = \begin{bmatrix} x - \frac{z_c}{g} \dddot{x} \\ y - \frac{z_c}{g} \dddot{y} \end{bmatrix} \quad (2)$$

where $g$ is the absolute value of gravity acceleration.

It is clear from (2) that the two elements of ZMP are very similar. For that, in the sequel, we only figure out how to build the controller for $p_x$.

To control $p_x$, Kajita et al [3] proposed to consider the cart table model which is shown in Fig. 1. It depicts a running cart of mass $m$ on a pedestal table whose mass is negligible.

2.1 Controlling ZMP

Let us define a new variable $u_x$ as the time derivative of $\dddot{x}$

$$u_x = \frac{d\dddot{x}}{dt} \quad (3)$$

The variable $u_x$ is usually called the jerk.

Regarding $u_x$ as the input of $p_x$, we can translate the equation of $p_x$ into the following dynamical
system

\[
\begin{align*}
\frac{d}{dt} \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} &= 
\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} + 
\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u_x \\
\end{align*}
\]

\( (4) \)

\[ p_x = \begin{bmatrix} 1 & 0 & -\frac{z_c}{g} \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} \]

As the humanoid robot is controlled in discrete time, we discretize the system \((4)\) with sampling time \(T_s\), the discrete control model for \(p_x\) can be expressed by the following formula

\[ z_{k+1} = Az_k + B\bar{u}_k \]
\[ p_k = Cz_k \]

\[ (5) \]

where

\[ z_k \triangleq \begin{bmatrix} x(kT_s) & \dot{x}(kT_s) & \ddot{x}(kT_s) \end{bmatrix}^T \]

\[ \bar{u}_k \triangleq u_x(kT_s) \]

\[ p_k \triangleq p_x(kT_s) \]

\[ (6) \]

\[ A = \begin{bmatrix} 1 & T_s & T_s^2/2 \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix}, 
B = \begin{bmatrix} T_s^3/6 \\ T_s^2/2 \\ T_s \end{bmatrix}, 
C = \begin{bmatrix} 1 & 0 & -\frac{z_c}{g} \end{bmatrix} \]

Regarding the similarity between the equations of \(p_x\) and \(p_y\), the above equation can be extended to
obtain the ZMP based control model as follows

\[
X_{k+1} = A_1 X_k + B_1 u_k \\
p_k = C_1 X_k
\]

where

\[
X_k \triangleq \begin{bmatrix} x(kT_s) \\ \dot{x}(kT_s) \\ \ddot{x}(kT_s) \\ y(kT_s) \\ \dot{y}(kT_s) \\ \ddot{y}(kT_s) \end{bmatrix}, \quad u_k \triangleq \begin{bmatrix} u_x(kT_s) \\ u_y(kT_s) \end{bmatrix}, \quad p_k \triangleq \begin{bmatrix} p_x(kT_s) \\ p_y(kT_s) \end{bmatrix}
\]

\[
A_1 = I_2 \otimes A, \quad B_1 = I_2 \otimes B, \quad C_1 = I_2 \otimes C
\]

\(I_2 \in \mathbb{R}^{2 \times 2}\) is the identity matrix, and \(\otimes\) is the operator of Kronecker product which is defined as follows:

\[
X \in \mathbb{R}^{m \times n}, \ Y \in \mathbb{R}^{p \times l}: \\
X \otimes Y \in \mathbb{R}^{mp \times nl} \triangleq \begin{bmatrix} x_{11}Y & \cdots & x_{1n}Y \\ \vdots & \ddots & \vdots \\ x_{m1}Y & \cdots & x_{mn}Y \end{bmatrix}
\]

where \(x_{ij}\) is the element of the \(i^{th}\) line and the \(j^{th}\) column of the matrix \(X\).

The above model has been used to generate a stable biped waking patterns using preview controller [3]. However, when the error between the output of this model and the real ZMP trajectory of the humanoid robot becomes bigger than the stability margin, then the robot falls down. The solution proposed by Kajita [3] is to inject this error in the preview controller as a second stage in order to eliminate the dynamic error. However, this procedure requires the dynamic simulation of the multi-body model of the humanoid robot, as a result it is a time consuming process.

To overcome this problem, we propose to model the ZMP of the humanoid robot by the model given in Figure 2. This model consists of two blocks, the first one is the previous cart-table model and the second one is a linear system with respect to the input \(u_k \otimes X_k\).

The obtained model is a quadratic system in the state space representation. This realization of quartic system is given by the following structure
\[ X_{k+1} = A_1 X_k + B_1 u_k \]
\[ Z_{k+1} = A_2 Z_k + B_2 (u_k \otimes X_k) \]
\[ y_{2,k} = C_2 Z_k \]
\[ p_k = C_1 X_k + C_2 Z_k \]

where \( u_k \in \mathbb{R}^2 \) is the input signal of the system expressed as in Eq. (8). \( p_k \in \mathbb{R}^2 \) is the output of the system (ZMP).

\{X_k, Z_k\} are the states, the dimension of \( X_k \) is equal to 6, and the dimension of \( Z_k \) is \( n \) that will be determined by the identification algorithm.

We have proven [12,14] that this class of nonlinear system is a special case of Volterra series of degree two. This class can play a useful role to model many real systems in practice.

### 3 System Identification

It is clear that the system is defined by the coefficient matrices of Structure (10). The unknown matrices are \( A_2, B_2 \) and \( C_2 \). That means that the structure parameters can be given by

\[ \theta = \begin{bmatrix} \text{vec}(A_2) \\ \text{vec}(B_2) \\ \text{vec}(C_2) \end{bmatrix} \]

where \( \text{vec}(\cdot) \) denotes the vectorization operator which is defined as follows

\[ \text{vec} : M \in \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^{m \cdot n} \]
\[ \text{vec}(M) = \text{vec} \begin{bmatrix} m_1 & m_2 & \cdots & m_n \end{bmatrix} = \begin{bmatrix} m_1^T & m_2^T & \cdots & m_n^T \end{bmatrix}^T \]
Given the input $u_k$ and the output $p_k$ of the real system, our goal is achieved if the output of the following model

$$
\begin{align*}
\hat{X}_{k+1} &= A_1 \hat{X}_k + B_1 u_k \\
\hat{Z}_{k+1}(\theta) &= A_2(\theta)\hat{Z}_k(\theta) + B_2(\theta)(u_k \otimes \hat{X}_k) \\
\hat{p}_k(\theta) &= C_1 \hat{X}_k + C_2(\theta)\hat{Z}_k(\theta)
\end{align*}
$$

approximates the output $p_k$ of the real system accurately enough.

This criterion can be transformed into the minimization of the output error with respect to the parameters $\theta$. Considering a data set of $N$ samples, the output-error cost function is given by

$$
J_N(\theta) = \frac{1}{N} \sum_{k=1}^{N} \left\| p_k - \hat{p}_k(\theta) \right\|_2^2 = \frac{1}{N} E_N(\theta)^T E_N(\theta) \tag{13}
$$

where $E_N(\theta) = [e(1)^T \ e(2)^T \ \cdots \ e(N)^T]^T$ is the error vector in which $e(k) = p_k - \hat{p}_k(\theta)$. The minimization of (13) is a nonlinear, nonconvex optimization problem. The numerical solution of this problem can be calculated by different algorithms. For instance, the iterative gradient search method is a popular one. This iterative method is based on the updating of the system parameters $\hat{\theta}$ as follows

$$
\hat{\theta}^i = \hat{\theta}^{i-1} - (\psi_N(\hat{\theta}^{i-1})\psi_N(\hat{\theta}^{i-1}) + \lambda^i I)^{-1}\psi_N^T(\hat{\theta}^{i-1}) E_N(\hat{\theta}^{i-1}) \tag{14}
$$

Where $\lambda^i$ is the regularization parameter and

$$
\psi_N(\theta) \triangleq \frac{\partial E_N(\theta)}{\partial \theta^T}
$$

is the Jacobian of the error vector $E_N(\theta)$.

### 3.1 Multiple Experiments Identification

In fact the model obtained by considering a single experiment might be not reliable [15]. This is because the input signals are not perfectly excited. To overcome this problem in identifying practical applications, we apply various sets of input signals. The objective of each set is exciting one or more modes of the system. The multiple experiments should be exploited simultaneously to obtain an accurate model of the system.

The optimization problem which consider $K$ experiments simultaneously can be formulated as follow

$$
J_K(\theta) = \frac{1}{K} \sum_{i=1}^{K} \frac{1}{N_i} \sum_{k=1}^{N_i} \left\| p_k^i - \hat{p}_k(\theta) \right\|_2^2 = \frac{1}{K} E_K(\theta)^T E_K(\theta) \tag{16}
$$
where
\[
E_K(\theta) = \left[ E_{N_1}^1(\theta)^T \ E_{N_2}^1(\theta)^T \cdots E_{N_K}^1(\theta)^T \right]^T
\] (17)
and
\[
E_{N_i}^i(\theta) = \frac{1}{\sqrt{N_i}} \left[ e^i(1)^T \ e^i(2)^T \cdots e^i(N_i)^T \right]^T
\] (18)
is the error vector in which \( e^i(k) = p_k^i - \hat{p}_k^i(\theta) \). \( p_k^i \) is the output of the system according to the input \( u_k^i \).

The estimated output \( \hat{p}_k^i(\theta) \) of the data set number \( i \) is given by the following model
\[
\hat{X}_{k+1}^i = A_1 \hat{X}_k^i + B_1 u_k^i
\]
\[
\hat{Z}_{k+1}^i(\theta) = A_2(\theta) \hat{Z}_k^i(\theta) + B_2(\theta)(u_k^i \otimes \hat{X}_k^i)
\]
\[
\hat{p}_k^i(\theta) = C_1 \hat{X}_k^i + C_2(\theta) \hat{Z}_k^i(\theta)
\] (19)
The minimization of (16) can be calculated, similarly to the case of single experiment, by using the iterative gradient search method as follows
\[
\hat{\theta}^{l+1} = \hat{\theta}^l - \left( \psi_K^T(\hat{\theta}^l) \psi_K(\hat{\theta}^l) + \lambda^l I \right)^{-1} \psi_K^T(\hat{\theta}^l) E_K(\hat{\theta}^l)
\] (20)
where
\[
\psi_K(\theta) = \begin{bmatrix}
\psi_{N_1}^1(\theta) \\
\psi_{N_2}^1(\theta) \\
\vdots \\
\psi_{N_K}^1(\theta)
\end{bmatrix}
\] (21)
and
\[
\psi_{N_i}^i(\theta) \triangleq \frac{\partial E_{N_i}^i(\theta)}{\partial \theta^T}
\] (22)
is the jacobian of the error vector \( E_{N_i}^i \).

In order to apply the identification algorithm, the input and output signals are supposed to be known. In our case, a sequence of a walking patterns have been deployed to identify the quadratic system. The input data sets have been calculated using the method proposed by Kajita et al [3] by defining the ZMP trajectories (the output signals).

3.2 Implementation Algorithm of the Identification method

The first step in applying the identification algorithm is determining the optimal dimension (\( n \)) of the state \( Z_k \). In order to achieve this task, a method based on oblique projection and subspace identification
method can be used, for more details please refer to [12, 14]. Another strategy, is to choose $n$ arbitrary then changing this value in order to obtain the quadratic system that fits accurately with the response of the real system.

Then, the implementation of system identification algorithm can be summarized as follows

1. Calculate the state $\hat{X}_i^k$, $\hat{Z}_i$ and $\hat{p}_k^i$ by simulating the system (19) with $\theta = \hat{\theta}^{-1}$.

2. Compute $E^i_N(\theta)$ using (18).

3. Calculate the matrix $\psi_K$ using (21, 22).

4. Calculate the update rule of the gradient search algorithm using (20), and obtain $\hat{\theta}^i$.

5. Perform the termination test for minimization. If true, the algorithm stops. Otherwise, return to Step 1. i.e. compute the values $J_L(\hat{\theta}^{-1})$ and $J_L(\hat{\theta}^i)$ using (16) and test if $\|J_L(\hat{\theta}^i) - J_L(\hat{\theta}^{-1})\|_2$ is small enough.

The above algorithm yields the matrices $A_2, B_2$ and $C_2$. Once the quadratic system is identified, not only the behavior of ZMP can be predicted using this model, but also the problem of ZMP servo tracking can be addressed as well.

### 4 Controller Design

The problem of designing a controller to follow a desired ZMP trajectory can be formulated as follows

$$
\min_{u_k} \sum_{k=0}^{N} u_k^T Q_u u_k + (p_k^{\text{ref}} - p_k)^T Q_e (p_k^{\text{ref}} - p_k)
$$

subject to

$$
X_{k+1} = A_1 X_k + B_1 u_k \\
Z_{k+1} = A_2 Z_k + B_2 (u_k \otimes X_k) \\
p_k = C_1 X_k + C_2 Z_k
$$

(23)

where $p_k^{\text{ref}}$ designs the desired ZMP trajectory. $N$ is the number of last sampling of the trajectory.

In this section we propose one algorithm for off-line controlling and two algorithms for on-line controlling of the quadratic system. The off-line algorithm is useful, for instance, in the case of motion planning. In this case, the path of the humanoid robot is provided off-line. However, for the reactive motions and human-robot interactions the on-line controlling algorithms are crucial.
4.1 Off-line walking pattern generation Algorithm

It is well known that the space of the admissible solutions of the minimization problem (23) is very large. In order to transform this space to a smaller dimensional space, we can use a basis of shape functions.

Let us consider a basis of shape functions \( S_k \) that is defined as follows

\[
S_k = \begin{bmatrix} S_{k1} & S_{k2} & \cdots & S_{kl} \end{bmatrix}^T
\]  

(24)

where \( S_{ki} \) denotes the value of shape function number \( i \) at the instant \( k \). The dimension of \( S_k \) is \( l \) which defines the dimension of the basis of shape functions.

The projection of \( u_k \) into the basis of shape functions \( S_k \) can be given by the following formula

\[
u_k = (I_2 \otimes S_k)^T u_S
\]

(25)

where \( I_2 \in \mathbb{R}^{2 \times 2} \) denotes the identity matrix, and \( u_S \in \mathbb{R}^{2l} \) denotes the vector of B-spline functions control points.

Thus, the optimization problem (23) can be rewritten as follows

\[
\min_{u_S} \sum_{k=0}^{L} u_S^T (I_2 \otimes S_k) Q_u (I_2 \otimes S_k)^T u_S + (p_{ref}^r - p_k)^T Q_e (p_{ref}^r - p_k)
\]

subject to

\[
X_{k+1} = A_1 X_k + B_1 (I_2 \otimes S_k)^T u_S
\]

\[
Z_{k+1} = A_2 Z_k + B_2 \left( (I_2 \otimes S_k)^T u_S \right) \otimes X_k
\]

\[
p_k = C_1 X_k + C_2 Z_k
\]

(26)

By defining the following variables

\[
J = \sum_{k=0}^{L} u_S^T (I_2 \otimes S_k) Q_u (I_2 \otimes S_k)^T u_S + (p_{ref}^r - p_k)^T Q_e (p_{ref}^r - p_k)
\]

\[
H = \begin{bmatrix}
X_{k+1} - A_1 X_k - B_1 (I_2 \otimes S_k)^T u_S \\
Z_{k+1} - A_2 Z_k - B_2 \left( (I_2 \otimes S_k)^T u_S \right) \otimes X_k \\
p_k - C_1 X_k - C_2 Z_k
\end{bmatrix}
\]

The optimization problem (26) is transformed into the following classical optimization problem
\[
\min_{u_S} J(u_S)
\]

subject to

\[H(u_S) = 0\]

(27)

In other words, the optimization problem has been transformed into finding the vector \(u_S \in \mathbb{R}^{2l}\). Note that the trajectory of CoM of the robot can be directly obtained from \(X_k\) by applying the input signal \(u_k\) to the quadratic system (10).

4.2 On-line Walking Pattern Generation Algorithm I

The idea of this algorithm is linearizing the quadratic system, then controlling the linearized system. The linearized system of quadratic system (10) around the value \(X_k = X^*\) is given by

\[
\begin{align*}
X_{k+1} &= A_1 X_k + B_1 u_k \\
Z_{k+1} &= A_2 Z_k + B_2 (u_k \otimes X^*) \\
\end{align*}
\]

(28)

Let us define the matrix \(B_{2,j}\) as follows

\[
B_{2,j} = B_2(:,(j-1)n_1+1:jn_1)
\]

(29)

where \(n_1\) denotes the dimension of \(X_k\) (in our case \(n_1 = 6\)), and the notation \(M(:,i:j)\) designs the sub-matrix of matrix \(M\) which contains the columns from \(i^{th}\) to \(j^{th}\) column.

Using (29) the variable \(B_2(u_k \otimes X^*)\) can be formulated as follows

\[
B_2(u_k \otimes X^*) = \begin{bmatrix} B_{2,1} X^* & B_{2,2} X^* & \cdots & B_{2,m} X^* \end{bmatrix} u_k \triangleq B^*_2 u_k
\]

(30)

Using the above definitions, the linearized system (28) can be reformulated as follows

\[
\begin{align*}
X_{k+1} &= AX_k + Bu_k \\
p_k &= CX_k \\
\end{align*}
\]

(31)
where
\[ X_k = \begin{bmatrix} X_k \\ Z_k \end{bmatrix}, \quad A = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}, \quad B = \begin{bmatrix} B_1 & 0 \\ 0 & B_2 \end{bmatrix}, \quad C = \begin{bmatrix} C_1 & C_2 \end{bmatrix} \] (32)

Therefore, the controller designing problem (23) is transformed into the following problem

\[
\min_{u_k} \sum_{k=0}^{L} u_k^T Q_u u_k + (p_k^{ref} - p_k)^T Q_e (p_k^{ref} - p_k)
\]

subject to
\[
X_{k+1} = AX_k + Bu_k \\
p_k = CX_k
\]

(33)

The above optimization problem is well known in automatic control as the preview control of a linear system. The optimal input signal \( u_k \) can be obtained by applying the classical Linear-quadratic state-feedback Regulator (LQR) technique. Thus, when the ZMP reference is previewed for \( N_L \) future steps at every sampling time, the optimal controller which minimizes the objective function (33) is given by

\[
u_k = -K X_k + \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_{N_L} \end{bmatrix} \begin{bmatrix} p_k^{ref} \\ p_{k+1}^{ref} \\ p_{k+2}^{ref} \\ \vdots \\ p_{k+N_L}^{ref} \end{bmatrix}
\] (34)

where \( K \) and \( f_i \) are calculated as

\[
K = \left( Q_u + B^T P B \right)^{-1} B^T P A \\
f_i = \left( Q_u + B^T P B \right)^{-1} B^T \left( A^T - K^T B^T \right)^{(i-1)} C^T Q_e
\] (35)

\( P \) is the solution of the following Riccati equation

\[
P = A^T P A + C^T Q_e C - A^T P B \left( R + B^T P B \right)^{-1} B^T P A
\] (36)

Note that the linearized system (28) is valid for a region around the value \( X_k = X^* \). Therefore a threshold \( \epsilon > 0 \) should be defined and the linearized system is updated with the new value of \( X_k \) if \( \| X_k - X^* \| > \epsilon \).

The trajectory of CoM can be obtained directly from \( X_k \) by simulating the quadratic system (10) using the control signal \( u_k \), which is the result of applying the above control algorithm.
4.3 On-line Walking Pattern Generation Algorithm II

The key idea of this algorithm can be summarized as follows

1. Controlling the quadratic system as a linear system. The approximated linear system is the linear subsystem (cart-table model) of the quadratic system.

2. Calculating the error between the dynamic simulation of the linear system and the quadratic system.

3. Correcting the error by considering it as perturbation, and calculate the appropriate input signal to decrease its effect.

Note that this algorithm works well when the dominant part of the system’s dynamic is the linear part. It is however the case of the walking motion.

Therefore, at first the optimization algorithm (23) is transformed into the following problem:

\[
\begin{align*}
\min_{u_k} & \sum_{k=0}^{L} u_k^T Q_u u_k + (p_k^{ref} - p_k)^T Q_e (p_k^{ref} - p_k) \\
\text{subject to} & \quad X_{k+1} = A_1 X_k + B_1 u_k \\
& \quad p_k = C_1 X_k
\end{align*}
\]  

(37)

Analogously to the previous controlling algorithm the above optimization problem can be solved using LQR technique.

The second step is to find the error between the outputs of the linear subsystem and the quadratic system by simulating the systems (37) and (10) using the optimal input signal \(u_k\), and obtain

\[
\Delta p_k^{ref} = C_2 Z_k
\]  

(38)

In order to decrease the effects of this error, one can considered it as an external perturbation, and apply LQR technique once more to calculate the variation of the input signal that should be added to the optimal \(u_k\) obtained from solving (37).

Thus, the new LQR problem can be formulated as follows
\[
\min_{u_k} \sum_{k=0}^{L} \Delta u_k^T Q_u \Delta u_k + (\Delta p_{k}\text{ref} - \Delta p_k)^T Q_{\Delta p} (\Delta p_{k}\text{ref} - \Delta p_k)
\]

subject to
\[
X_{k+1} = A_1 X_k + B_1 \Delta u_k
\]
\[
\Delta p_k = C_1 X_k
\]

Analogously to (37) the variation of the input signal $\Delta u_k$ can be obtained. As a result the input signal that should be applied to the humanoid robot becomes $u_k^* = u_k + \Delta u_k$.

5 Experimental Results

5.1 Humanoid Robot: Kinematic Structure

The proposed walking patterns generation algorithms have been validated using the cybernetic human HRP-4C [13]. HRP-4C is a life-size humanoid robot which was developed for the entertainment use such as a fashion model, master of ceremony of various events and so on. For these applications, it has been decided to make it more humanlike than the humanoid robots that have been developed so far [16,17]. In order to realize a humanlike shape, its dimensions are designed referring to a database of Japanese women of 20th century [18]. The basic joint configurations are shown in Figure 3 (right) (Face and hand joints are not displayed).

Figure 3: Exterior and joint configuration of cybernetic human HRP-4C (Face and hand joints are not displayed).
Table 1: Specifications of the cybernetic human HRP-4C

<table>
<thead>
<tr>
<th>degrees of freedom</th>
<th>arm</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>hand</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>leg</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>waist</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>neck</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>face</td>
<td>8</td>
</tr>
<tr>
<td>height</td>
<td>158 cm</td>
<td></td>
</tr>
<tr>
<td>weight (including batteries)</td>
<td>43 kg</td>
<td></td>
</tr>
<tr>
<td>sensors</td>
<td>force/torque sensor×2, inertia sensor</td>
<td></td>
</tr>
</tbody>
</table>

5.2 Identification of Quadratic System

In order to identify the quadratic system, we have simulated several walking patterns with different step length and walking speed values. We define the model accuracy as the Percent Variance Accounted For (%VAF)

\[
\%VAF \triangleq \left(1 - \frac{\sum_{k=1}^{L} (p_k - \hat{p}_k)^T (p_k - \hat{p}_k)}{\sum_{k=1}^{L} (p_k - \bar{p}_k)^T (p_k - \bar{p}_k)}\right) \times 100
\]

where \(\hat{p}_k\) denotes the estimated output signal (the output of the quadratic system), and \(\bar{p}_k\) is the mean value of the real output of the system \(p_k\) (ZMP trajectory of the multi-body model).

The optimal dimension of the state \(Z_k\) has been calculated according to the procedure explained in [14], and we obtained \(n = 8\).

In order to validate the identified model, the number of conducted simulation walking patterns and the accuracy of the identified quadratic system model are given in Table 2.

Table 2: Accuracy of the identified model

<table>
<thead>
<tr>
<th>Number of experiments</th>
<th>Accuracy of the proposed model (%VAF)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>89.7</td>
</tr>
<tr>
<td>5</td>
<td>93.1</td>
</tr>
<tr>
<td>10</td>
<td>97.2</td>
</tr>
</tbody>
</table>

Depending on the accuracy reported in Table 2, we have chosen the identified model which is the result of using 10 walking patterns. In Figure 5, we observe that the quadratic system is perfectly able to capture the dynamic difference between the multi-body model and the simple cart-table model. For more details on the identification procedure please refer to [19].
Figure 4: The reference ZMP trajectory (dashed lines), the ZMP of multi-body model (solid lines) and the safety region of stability (dotted lines).

Figure 5: The ZMP error between the multi-body model of humanoid robot and the simple cart-table model (solid line), and the output of the second block of the quadratic system (dashed line).

5.3 Results: Off-line Walking Pattern Generation Algorithm

The application of the off-line walking pattern generation algorithm in order to track a desired trajectory of ZMP is applied to the cybernetic human HRP4-C. We have chosen a basis $(S_k)$ of 40 B-spline functions. The optimization algorithm stops when the norm of the error between the planned ZMP ($p^{ref}$) and the output of quadratic system is less than $10^{-4} \ m$. The ZMP error between the multi-body model of humanoid robot and the planned ZMP trajectory ($p^{ref}$) is given in Figure 6.

From Figure 6 we can conclude the following remarks:
1. The planned ZMP trajectory is well tracked by the proposed controller.

2. The error between the ZMP of multi-body model of humanoid robot and the planned ZMP trajectory \( p_{x}^{ref} \) is small enough (less than 2 cm), and the ZMP stays always inside of the region of stability. Moreover, that proves that the quadratic system is an accurate model of the walking motion.

Figure 6: Offline walking pattern generation algorithm: the ZMP error between the multi-body model of humanoid robot and the planned ZMP trajectory \( p_{x}^{ref} \) (solid line), and the safety margin of stability (dashed line).

In order to validate the obtained results, we have simulated the generated motion using the dynamical simulator OpenHRP3 [20,21]. Snapshots of the simulated walking pattern are given in Figure 7.

Figure 7: Off-line walking pattern generation algorithm: snapshots of the simulated walking motion
5.4 Results: on-line Walking Pattern Generation Algorithm I

In this experiment, we apply the on-line walking pattern generation algorithm I to generate the CoM trajectory in order to track a desired trajectory of ZMP. The ZMP error between the multi-body model of humanoid robot and the planned ZMP trajectory ($p_{\text{ref}}$) is given in Figure 8.

![Graph](image)

Figure 8: On-line walking pattern generation algorithm I: the ZMP error between the multi-body model of humanoid robot and the planned ZMP trajectory ($p_{\text{ref}}$) (solid line), and the safety margin of stability (dashed line).

From Figure 8, we observe that the ZMP trajectory is always inside of the safety region of stability. However, the ZMP tracking error is bigger than that of off-line walking pattern generation. On the other hand, this controller can be applied on-line on account of its low computational complexity.

Snapshots of the real experiment conducted on the cybernetic human HRP-4C are given in Figure 9.

![Snapshots](image)

Figure 9: On-line walking pattern generation algorithm I: Snapshots of the real experiment using the cybernetic human HRP-4C.
5.5 Results: on-line Walking Pattern Generation Algorithm II

Analogously to the previous case, we apply the on-line walking pattern generation algorithm II to generate the CoM trajectory in order to track a desired trajectory of ZMP. The ZMP error between the multi-body model of humanoid robot and the planned ZMP trajectory ($p_{ref}^{err}$) is given in Figure 10.

![Figure 10: On-line walking pattern generation algorithm II: the ZMP error between the multi-body model of humanoid robot and the planned ZMP trajectory ($p_{ref}^{err}$) (solid line), and the safety margin of stability (dashed line).]

From Figure 10, we observe that the ZMP tracking error becomes bigger. However, the ZMP is always inside of the safety region of stability, and the dynamical stability of the walking patterns has been verified using the dynamical simulator OpenHRP3.

6 Conclusion and Discussion

In this paper, we proposed a new nonlinear model in order to approximate the dynamic behavior of humanoid robot walking motion. The linear part of the nonlinear model is the cart-table model proposed in [3]. The nonlinear model is a quadratic system which is a special case of Volterra series. The main advantage of this model is not only predicting the ZMP trajectory of the humanoid robot without the calculation of the multi-body dynamics, but also providing the trajectory of Centre of Mass (CoM) of the humanoid robot in order to track a planned ZMP trajectory.

To identify the quadratic system, we proposed an adapted identification algorithm. This algorithm is based on using multiple walking patterns in order to identify an accurate model. Moreover, the problem of controller design of quadratic system has been investigated as well. We proposed on-line and off-line walking pattern generation algorithms based on controlling the quadratic system. Even-though the off-line walking pattern generation algorithm can track the planned ZMP trajectory precisely, the computational time is not adequate for real-time applications. On the other hand, the on-line walking
pattern generation algorithms are devoted for real-time applications and the ZMP trajectory is tracked as much as possible. The conducted simulations using the dynamical simulator OpenHRP3 [20,21] and the real experiments using the cybernetic human HRP4-C [13] have pointed out that the ZMP trajectory stays always inside of the safety region of stability, and the generated walking patterns are dynamically stable.

The actual limitations of the proposed model is that when the walking trajectories are curved, the quadratic system becomes not accurate enough to capture the ZMP behavior of the humanoid robot. As a result, the generated CoM trajectory might yield an unstable walking motion and the robot might fall down. One reason of this phenomenon is that the input signal of the actual model includes only the jerk functions of the horizontal projections of the CoM. One might think that including the rotational velocities of CoM in the input signal might yield more accurate model. This point will be investigated in future work.

Finally, the first obtained results of this challenging problem are encouraging, enhancing therefore the results to be more robust and accurate is the intent of our future work.

7 Acknowledgment

This research was partially supported by a Grant-in-Aid for Scientific Research from the Japan Society for the Promotion of Science (20-08816).

REFERENCES


