

# A Model Predictive Control Approach for the Partner Ballroom Dance Robot

Gabriele Buondonno\* Federico Patota\* Hongbo Wang<sup>o</sup> Alessandro De Luca\* Kazuhiro Kosuge<sup>o</sup>

**Abstract**—A model predictive controller is developed for following the position of a human dancer in robot ballroom dancing. The control design uses a dynamic model of a dancer, based on a variant of the so-called 3D Linear Inverted Pendulum Mode that includes also the swing foot. This model serves as a basis for a Kalman predictor of the human motion during the single-support phase, while a simpler kinematic technique is used during the double-support phase. The output of the prediction filter enables to design a Model Predictive Control (MPC) law, by recursively solving on line and within a preview window a convex linear-quadratic optimization problem, constrained by differential kinematic bounds on robot commands. Two different control strategies, either at the velocity or at the acceleration level, are proposed and compared in simulations and in actual experiments. Accurate and reactive behaviors are obtained by the ballroom robot follower, confirming the benefit of the predictive/filtering nature of a MPC approach to handle uncertainty of human intentions and noisy signals.

## I. INTRODUCTION

There is a growing interest in handling situations of physical Human-Robot Interaction (pHRI) where the human motion intention needs to be monitored and the robot has to react on line in order to achieve natural and successful collaboration. Recently, waltz dancing has been considered as a mean for studying pHRI in leader-follower modality. In waltz, the roles of the two dancers are clearly defined: the robot partner acts as a follower and should track the leading human dancer. During ballroom dancing, the movements and transitions of steps are particularly smooth, which makes the control problem simpler. In addition, several constraints are imposed on the posture of the leader and this may simplify the modeling effort. For instance, the upper body has to be rigid, straight, and vertical, and the swing foot should never be raised too much from the ground.

The problem of pHRI during waltz was first addressed in [1]–[3]. In these papers, the human dynamics in the sagittal plane during the single-support phase was studied by approximating it with a linear inverted pendulum model, as introduced in [4] for biped locomotion. The dance robot was instructed to emulate a similar dynamics, while interaction forces between the two dancers were modeled by a virtual spring-damper connection. The coupled dynamics of the resulting system was then used to design the robot motion controller.

\* The authors are with the Dipartimento di Ingegneria Informatica, Automatica e Gestionale, Sapienza Università di Roma, Via Ariosto 25, 00185, Rome, Italy. Email: {buondonno, patota, deluca}@diag.uniroma1.it

<sup>o</sup> The authors are with the Department of Bioengineering and Robotics, Tohoku University, 6-6-01, Aoba, Aramaki, Aoba-ku, Sendai 980-8579, Japan. Email: {h.wang, kosuge}@irs.mech.tohoku.ac.jp

These previous works were mainly focused on the translational control of the dance robot along a line. In this paper, we extend the study and consider a more complete motion of the Partner Ballroom Dance Robot (PBDR), including both translation and rotation on the floor plane. In our modeling, we conveniently resort to the 3D Linear Inverted Pendulum Mode (3D-LIPM) for biped dynamics [5], with the addition of a swing foot, as introduced by [6], and considering a simplifying assumption. The resulting dynamic model takes the more handy form of a bilinear system [7], with only products of states and inputs. Using this model, we can obtain a prediction of the planar pose of the human dancer over a suitable time window. After discretization, the problem of following accurately the human dancer is formulated as a constrained linear-quadratic (LQ) optimization problem, under velocity, acceleration, and jerk bounds on the commands.

The availability of a nominal future motion of the human leader, according to the smooth sequence of waltz steps and as predicted by the approximate model, and the need for compensating actual motion uncertainties/deviations and noisy sensor measurements, both suggest the use of a Model Predictive Control (MPC) framework for the design of a controller that will let the PBDR follow the human planar pose—an original application of MPC in this context. An extensive review of the existing approaches and perspectives for MPC can be found in [8], [9].

After detailing the model of a human dancer during a single-support phase and its discretization (Sect. II), the related estimation/prediction structure of an Extended Kalman Filter (EKF) is presented in Sect. III. Two MPC strategies, at the velocity or at the acceleration control level, are proposed in Sect. IV. Sections V and VI compare the obtained performance, respectively in simulations and in actual experiments on the Tohoku PBDR.

## II. DANCER MODELING

### A. 3D LIPM with swing foot

The dynamic model of a biped robot during a single-support phase presented in [6] consists of a 3D-LIPM with the addition of a swing foot. This model was originally intended for controlling a humanoid in the execution of quick turning maneuvers. Our claim is that this model is accurate enough to be used also for estimating the state of a human dancer and predicting his/her next movements. The original model is further simplified by making a simple consideration regarding the waltz dance: since the swing foot should never

be raised too much in this case, the height  $z_s$  of the swing foot is considered here to be constant (and equal to zero).

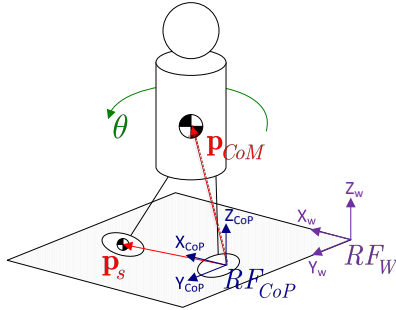


Fig. 1. The 3D-LIPM with swing foot, where  $\mathbf{p}_{CoM} = (x \ y \ z)^T$  is the position of the center of mass,  $\mathbf{p}_s = (x_s \ y_s \ z_s)^T$  is the position of the swing foot, and  $\theta$  is the absolute orientation of the human dancer. Vectors are expressed relative to the reference frame  $RF_{CoP}$ .

We define the system state vector as

$$\mathbf{x}^T = (x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6) = (x \ \dot{x} \ y \ \dot{y} \ \theta \ \dot{\theta}), \quad (1)$$

where  $x$  and  $y$  are the coordinates of the position of the center of mass (CoM) relative to the center of pressure (CoP), and  $\theta$  is the absolute orientation of the human dancer. The global reference frame  $RF_W$  is parallel to  $RF_{CoP}$ , i.e., the frame attached to the CoP. The resulting dynamic model is

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{g}{z_c} x_1 + \frac{g}{z_c} \frac{m_s}{m} x_5 + \frac{F_x}{m} \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= \frac{g}{z_c} x_3 + \frac{g}{z_c} \frac{m_s}{m} y_5 + \frac{F_y}{m} \\ \dot{x}_5 &= x_6 \\ \dot{x}_6 &= \frac{g}{z_c} \frac{m_s}{I} x_5 x_3 - \frac{g}{z_c} \frac{m_s}{I} y_5 x_1 + \frac{m_s}{I} \alpha_5 + \frac{\tau_z}{I}, \end{aligned} \quad (2)$$

where

- $m$  and  $I$  are the main body mass and inertia
- $z_c$  is the constant value of the CoM height
- $m_s$  is the swing foot mass
- $x_s$  and  $y_s$  are the swing foot coordinates ( $z_s = 0$ )
- $g$  is the gravity acceleration
- $F_x$ ,  $F_y$ , and  $\tau_z$  are the interaction forces and torque
- $\alpha_5 = (y_s \ddot{x}_s - x_s \ddot{y}_s)$ .

The  $x_s$  and  $y_s$  coordinates are measured using laser range finders (LRFs) placed on the PBDR. Derivation of the model employed can be found in [6]. Simulations have shown that, in waltz,  $\alpha_5$  in (2) can (and will) be neglected. All other parameters are constant and should be chosen so to represent the actual human dancer. As a result, the human dynamics in SSP can be expressed as a bilinear system of the form

$$\begin{aligned} \dot{\mathbf{x}} &= A(\mathbf{m})\mathbf{x} + B_m \mathbf{m} = A(\mathbf{p}_s)\mathbf{x} + B \mathbf{p}_s + \bar{B} \mathbf{f} \\ \mathbf{m}^T &= (x_s \ y_s \ | \ F_x \ F_y \ \tau_z) = (\mathbf{p}_s^T \ | \ \mathbf{f}^T), \end{aligned} \quad (3)$$

in which a product of state  $\mathbf{x}$  and input  $\mathbf{m}$  appears as the only nonlinearity in the equations. Since a measurement of interaction force/torque was not used, we have set  $\mathbf{f} = \mathbf{0}$  in simulations and experiments (only the input matrix  $B$  will be used in the following). Indeed, if the robot follows the human dancer well, the interaction forces between the two partners should remain negligible.

## B. Discretization of the model

In order to implement a Kalman estimator and a predictor, a discretized version of the continuous system (3) is required. The input vector  $\mathbf{m}$  is assumed to be constant during each sampling interval  $\Delta t$ , and so matrix  $A(\mathbf{m})$  becomes constant over this interval. This allows resorting to standard tools for the *exact* discretization of linear systems also in the case of the bilinear system (3). After discretization, we obtain

$$\mathbf{x}_{k+1} = g(\mathbf{x}_k, \mathbf{m}_k) = A_d(\mathbf{p}_{s,k}) \mathbf{x}_k + B_d(\mathbf{p}_{s,k}) \mathbf{p}_{s,k}, \quad (4)$$

with

$$A_d = \begin{pmatrix} c_h & a_1 s_h & 0 & 0 & 0 & 0 \\ s_h/a_1 & c_h & 0 & 0 & 0 & 0 \\ 0 & 0 & c_h & a_1 s_h & 0 & 0 \\ 0 & 0 & s_h/a_1 & c_h & 0 & 0 \\ a_2 y_{s,k} & a_3 y_{s,k} & -a_2 x_{s,k} & -a_3 x_{s,k} & 1 & \Delta t \\ a_4 y_{s,k} & a_2 y_{s,k} & -a_4 x_{s,k} & -a_2 x_{s,k} & 0 & 1 \end{pmatrix}$$

$$B_d = \begin{pmatrix} b_1 & 0 \\ b_2 & 0 \\ 0 & b_1 \\ 0 & b_2 \\ b_3 y_{s,k} & -b_3 x_{s,k} \\ -b_4 y_{s,k} & b_4 x_{s,k} \end{pmatrix}$$

and where

$$\begin{aligned} s_h &= \sinh\left(\frac{\Delta t}{a_1}\right), \quad c_h = \cosh\left(\frac{\Delta t}{a_1}\right) \\ a_1 &= \sqrt{\frac{z_c}{g}}, \quad a_2 = \frac{m_s(1-s_h)}{I}, \quad a_3 = \frac{m_s(\Delta t - a_1 s_h)}{I}, \quad a_4 = -\frac{m_s s_h}{a_1 I} \\ b_1 &= \frac{m_s(c_h - 1)}{m}, \quad b_2 = \frac{m_s s_h}{m a_1} \\ b_3 &= \frac{m_s^2}{m I} \left(\frac{\Delta t^2}{2a_1^2} + 1 - c_h\right), \quad b_4 = \frac{g m_s^2}{z_c I m} (s_h a_1 - \Delta t). \end{aligned}$$

We remark that the resulting discretized system (4) is no longer linear in the input. Moreover, since everything is expressed in the human CoP coordinate frame, when using this model one needs to perform a conversion from the global frame to the local one and vice versa.

## III. ESTIMATION AND PREDICTION

We develop an estimator and a predictor of the state of the human dancer. A Kalman estimator will work only during the single-support phase (SSP), while during the double-support phase (DSP) a simpler low-pass filtering of measured data will be performed. Similarly, the predictor will operate in two different modalities: model (4) will be used during a SSP, while a model-free prediction is made during a DSP.

### A. Estimator

For state estimation, a standard Extended Kalman Filter (EKF) has been chosen. From (4), we compute the state transition matrix and the observation matrix associated with the EKF (see, e.g., [10]) respectively as

$$G_k = \frac{\partial g(\mathbf{x}_k, \mathbf{p}_{s,k})}{\partial \mathbf{x}_k} = A_d(\mathbf{p}_{s,k}), \quad H_k = I_{6 \times 6}. \quad (5)$$

The process noise (diagonal) covariance matrix was obtained numerically using motion capture (MoCap) data. MoCap

data were recorded from a continuous run of human dancer, sampled every  $\Delta t = 10$  ms. These data were considered as the ground truth and used at each time step for predicting the next state by (4) during SSP. The computed RMS errors were then taken as standard deviations of the state components of this noisy process, as reported in Tab. I.

TABLE I  
STANDARD DEVIATION OF THE STATE

| $\sigma_x$ [m] | $\sigma_{\dot{x}}$ [m/s] | $\sigma_y$ [m] | $\sigma_{\dot{y}}$ [m/s] | $\sigma_\theta$ [deg] | $\sigma_{\dot{\theta}}$ [deg/s] |
|----------------|--------------------------|----------------|--------------------------|-----------------------|---------------------------------|
| 0.0040         | 0.0216                   | 0.0016         | 0.0198                   | 4.1196                | 10.3591                         |

### B. Predictor

Given  $\mathbf{x}$  and  $\Delta t$ , the goal of this module is to predict the human state for a suitable number  $N$  of time samples ahead (prediction interval). During a SSP, two assumptions were made:

- 1) The human dancer will remain in single-support phase during the prediction interval.
- 2) The swing foot velocity remains constant in the prediction interval.

The prediction is then obtained simply by the repeated application for  $N$  times of eq. (4). When designing the predictor, the most critical quantity is the size of the prediction time window  $T = N \cdot \Delta t$ . The tradeoff is between a large enough window to make the prediction meaningful and an excessive size that will result in a loss of accuracy at instants that are further away in time.

TABLE II  
RMSE OF THE PREDICTION FOR DIFFERENT TIME WINDOWS

|                   | $T = 0.1$ s | $T = 0.2$ s | $T = 0.3$ s |
|-------------------|-------------|-------------|-------------|
| Position [m]      | 0.0101      | 0.0416      | 0.0994      |
| Orientation [deg] | 1.5413      | 6.0275      | 13.688      |

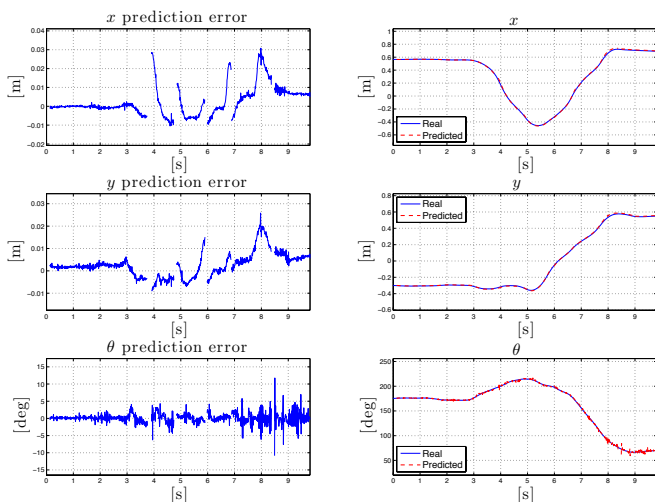


Fig. 2. Performance of the predictor during SSP for  $T = 0.1$  s. Prediction errors [left] and comparison of ground truth and estimated values [right]. The discontinuities in the plots represent DSP intervals.

Table II shows how the root mean squared errors (RMSE) of the prediction increases with the size  $T$  of the prediction window along a typical test trajectory. The errors are still comparable to the process noise for  $T = 0.1$  s, and get worse for larger time windows. After a number of trial-and-error simulations, we found that in our case a value  $T = 0.1$  s is the most convenient one. Performance of the predictor during SSP along the same test trajectory is shown in Fig. 2.

During DSP, a simpler prediction approach is followed. The linear and angular velocities of the human are computed using the sensor data obtained from the LRFs, and are assumed to be constant during the whole prediction window. Thus, the predicted planar pose of the human dancer at the  $j$ -th sampling instant in the future is obtained by simply applying these velocities to the current human pose for a time  $j\Delta t$ .

## IV. MODEL PREDICTIVE CONTROL

The basic idea of MPC is to compute, at each sampling instant, a full sequence of inputs so as to minimize a performance measure over a given time window. However, only the first element of this input sequence is applied, and then the whole computation is repeated at the next iteration in time. Comparing this to a purely reactive controller, a MPC takes actions that may well be not instantaneously/locally optimal, but that will perform certainly better in the long run. Moreover, kinematic constraints are easily taken into account in MPC and the information provided by model-based predictions can be fully exploited.

In the following, we characterize the robot configuration as a planar frame, directly controlled by its linear and angular velocity or acceleration (the wheeled mobile base of the Tohoku PBDR is in fact omnidirectional). The control goal is to track a frame placed at a given distance  $d$  in front of the human dancer, and having his/her same orientation. Define:

- $\mathbf{p}_{r,k} = (x_{r,k}, y_{r,k}, \theta_{r,k})^T$ , the robot pose at instant  $k$ ;
- $\mathbf{p}_{h,k} = (x_{h,k}, y_{h,k}, \theta_{h,k})^T$ , the planar pose of the tracked frame in front of the human dancer at instant  $k$ ;
- $\mathbf{u}_k = (u_{x,k}, u_{y,k}, u_{\theta,k})^T$ , the control input at instant  $k$ ;
- $N$ , the number of samples in the preview horizon.

### A. Problem formulation

At the  $k$ -th time instant, with a preview of  $N$  steps, the goal is to minimize the cost function

$$J_k(\bar{\mathbf{u}}) = \frac{1}{2} \sum_{j=1}^N \mathbf{e}_{k+j}^T(\bar{\mathbf{u}}) W_j \mathbf{e}_{k+j}(\bar{\mathbf{u}}), \quad (6)$$

where  $\bar{\mathbf{u}} = (\mathbf{u}_k^T, \dots, \mathbf{u}_{k+N-1}^T)^T \in \mathbb{R}^{3N}$  is the input command and  $\mathbf{e}_k = \mathbf{p}_{h,k} - \mathbf{p}_{r,k}$  is the error vector, subject to maximum velocity and acceleration constraints. The weights  $W_j$  are  $3 \times 3$  positive definite matrices.

In general,  $\mathbf{p}_{h,k+j}$ , for  $j = 1, \dots, N$ , will be obtained by the predictor. The robot pose  $\mathbf{p}_{r,k+1}$  can be expressed as:

$$\mathbf{p}_{r,k+j} = \mathbf{p}_{r,k} + f(\mathbf{u}_k, \dots, \mathbf{u}_{k+j-1}). \quad (7)$$

Defining the two vectors  $\bar{\mathbf{p}}_h = (\mathbf{p}_{h,k+1}^T, \dots, \mathbf{p}_{h,k+N}^T)^T \in \mathbb{R}^{3N}$  and  $\bar{\mathbf{p}}_r(\bar{\mathbf{u}}) = (\mathbf{p}_{r,k+1}^T, \dots, \mathbf{p}_{r,k+N}^T)^T \in \mathbb{R}^{3N}$ , as well as the matrix  $W = \text{blockdiag}\{W_1, \dots, W_N\} \in \mathbb{R}^{3N \times 3N}$ , the problem is reduced to:

$$\min_{\bar{\mathbf{u}} \in \mathcal{U}} \frac{1}{2} (\bar{\mathbf{p}}_h - \bar{\mathbf{p}}_r(\bar{\mathbf{u}}))^T W (\bar{\mathbf{p}}_h - \bar{\mathbf{p}}_r(\bar{\mathbf{u}})) \quad (8)$$

where  $\mathcal{U}$  is the admissible set of the control input variables. We will show in the following how it is possible to reduce (8) to a standard LQ optimization problem, respectively using velocity or acceleration control inputs. From now on, dependency from  $\bar{\mathbf{u}}$  will be omitted for notation simplicity.

### B. Velocity-level control design

In this case, the discretized robot dynamics is

$$\mathbf{p}_{r,k+1} = \mathbf{p}_{r,k} + \mathbf{u}_k \Delta t. \quad (9)$$

Thus, the dependency of the robot pose at time instant  $k+i$  from the previous inputs is

$$\mathbf{p}_{r,k+i} = \mathbf{p}_{r,k} + \sum_{j=1}^{i-1} \mathbf{u}_{r,k+j} \Delta t, \quad (10)$$

for  $1 \leq i \leq N$ . We can write

$$\bar{\mathbf{p}}_r = (\mathbf{p}_{r,k}^T, \dots, \mathbf{p}_{r,k}^T)^T + C \bar{\mathbf{u}}, \quad (11)$$

where

$$C = \Delta t \begin{pmatrix} I_{3 \times 3} & & & \\ & \ddots & & \\ & & \ddots & \\ I_{3 \times 3} & \dots & I_{3 \times 3} & \end{pmatrix} \in \mathbb{R}^{3N \times 3N}. \quad (12)$$

Thus, defining

$$\mathbf{r} = \bar{\mathbf{p}}_h - (\mathbf{p}_{r,k}^T, \dots, \mathbf{p}_{r,k}^T)^T \quad (13)$$

and replacing (11) and (13) in (8), we obtain

$$\min_{\bar{\mathbf{u}} \in \mathcal{U}} \frac{1}{2} (\mathbf{r} - C \bar{\mathbf{u}})^T W (\mathbf{r} - C \bar{\mathbf{u}}), \quad (14)$$

which can be expressed as the quadratic criterion

$$\min_{\bar{\mathbf{u}} \in \mathcal{U}} \mathbf{h}^T \bar{\mathbf{u}} + \frac{1}{2} \bar{\mathbf{u}}^T \tilde{W} \bar{\mathbf{u}} \quad (15)$$

where  $\mathbf{h} = -C^T W \mathbf{r}$  and  $\tilde{W} = C^T W C$ .

To complete the LQ formulation, we introduce the bounds on kinematic quantities, leading to linear constraints on the commands<sup>1</sup>:

- *velocity bounds*

$$\forall i \in [0, N-1], \quad |\mathbf{u}_{k+i}| \leq \mathbf{v}_{max}; \quad (16)$$

- *acceleration bounds*

$$\forall i \in [0, N-1], \quad |\mathbf{u}_{k+i} - \mathbf{u}_{k+i-1}| \leq \mathbf{a}_{max} \Delta t. \quad (17)$$

In order to avoid excessive acceleration oscillations, also *jerk bounds* can be added, given by:

$$\forall i \in [0, N-1] \quad |\mathbf{u}_{k+i} - 2\mathbf{u}_{k+i-1} + \mathbf{u}_{k+i-2}| \leq \mathbf{j}_{max} \Delta t^2. \quad (18)$$

<sup>1</sup>In the following, for a vector  $\mathbf{u}$ , we denote by  $|\mathbf{u}|$  its component-wise absolute value. Similarly, with  $\mathbf{a} \leq \mathbf{b}$ , where both  $\mathbf{a}$  and  $\mathbf{b}$  are vectors, we denote an inequality which should hold component-wise.

All these constraints can be easily expressed in matrix form as  $A \bar{\mathbf{u}} \leq \mathbf{b}$ . Note also that the formulated problem is always strictly convex. This is because (16), (17), and (18) constitute a convex polytope and  $\tilde{W} = C^T W C$  is always positive definite.

### C. Acceleration-level control design

In this case, the discretized robot dynamics is

$$\begin{aligned} \mathbf{p}_{r,k+1} &= \mathbf{p}_{r,k} + \mathbf{v}_{r,k} \Delta t + \frac{1}{2} \mathbf{u}_k \Delta t^2 \\ \mathbf{v}_{r,k+1} &= \mathbf{v}_{r,k} + \mathbf{u}_k \Delta t \end{aligned} \quad (19)$$

where  $\mathbf{v}_{r,k}$  is the linear and angular velocity vector of the robot, and is part of its state at time instant  $k$ . This leads to

$$\mathbf{p}_{r,k+i} = \mathbf{p}_{r,k} + \mathbf{v}_{r,k} i \Delta t + \sum_{j=0}^{i-1} \left( i - j - \frac{1}{2} \right) \mathbf{u}_{k+j} \Delta t^2, \quad (20)$$

for  $1 \leq i \leq N$ . We can write

$$\begin{aligned} \bar{\mathbf{p}}_r &= (\mathbf{p}_{r,k}^T, \mathbf{p}_{r,k}^T, \dots, \mathbf{p}_{r,k}^T)^T \\ &+ \left( \mathbf{v}_{r,k}^T, 2 \mathbf{v}_{r,k}^T, \dots, N \mathbf{v}_{r,k}^T \right)^T \Delta t + C \bar{\mathbf{u}}. \end{aligned} \quad (21)$$

In this case, matrix  $C$  preserves a block lower triangular structure, in which the  $3 \times 3$  block  $C_{ij}$  at the  $i$ -th block row and  $j$ -th block column has the expression

$$C_{ij} = I_{3 \times 3} \cdot \left( i - j + \frac{1}{2} \right) \Delta t^2. \quad (22)$$

Vector  $\mathbf{r}$  is now defined as

$$\begin{aligned} \mathbf{r} &= \bar{\mathbf{p}}_h - \left( \mathbf{p}_{r,k}^T, \mathbf{p}_{r,k}^T, \dots, \mathbf{p}_{r,k}^T \right)^T \\ &- \left( \mathbf{v}_{r,k}^T, 2 \mathbf{v}_{r,k}^T, \dots, N \mathbf{v}_{r,k}^T \right)^T \Delta t. \end{aligned} \quad (23)$$

In this way, the problem can still be expressed as

$$\min_{\bar{\mathbf{u}} \in \mathcal{U}} \mathbf{h}^T \bar{\mathbf{u}} + \frac{1}{2} \bar{\mathbf{u}}^T \tilde{W} \bar{\mathbf{u}} \quad (24)$$

which is formally identical to (15), now with  $\mathbf{h}$  and  $\tilde{W}$  built using the newly defined  $C$  and  $\mathbf{r}$  in (22) and (23). As in the velocity control case, we have two main types of constraints:

- *velocity bounds*

$$\forall i \in [1, N], \quad |\mathbf{v}_{r,k} + \sum_{j=0}^{i-1} \mathbf{u}_{k+j} \Delta t| \leq \mathbf{v}_{max}; \quad (25)$$

- *acceleration bounds*

$$\forall i \in [0, N-1], \quad |\mathbf{u}_{k+i}| \leq \mathbf{a}_{max}. \quad (26)$$

For limiting acceleration oscillations, *jerk bounds* can also be added:

$$\forall i \in [0, N-1] \quad |\mathbf{u}_{k+i} - \mathbf{u}_{k+i-1}| \leq \mathbf{j}_{max} \Delta t \quad (27)$$

As before, all constraints can be converted to a linear inequality form  $A \bar{\mathbf{u}} \leq \mathbf{b}$ . The same previous considerations about problem convexity hold true.

## V. SIMULATION RESULTS

In simulation, the controller was given a human waltz step trajectory, recorded by MoCap, as reference motion input for execution. Single-support phases were distinguished from (typically short) double-support phases with the aid of an algorithm based on velocity thresholds. The parameters used in the simulations are given in Tabs. III–IV. In order to solve the minimization problem, any generic LQ solver can be used. In our implementation, we employed an interior-point algorithm for convex LQ problems [11]. The weighting matrices in (6) were chosen as  $W_j = (20/2^j) \cdot I_{3 \times 3}$ . In all plots, the control action begins at the dotted (green) vertical line. Kinematic bounds are represented by horizontal dotted (black) lines.

TABLE III  
PREDICTOR PARAMETERS

| $m$ [Kg] | $m_s$ [Kg] | $I$ [Kg·m <sup>2</sup> ] | $d$ [m] | $N$ | $\Delta t$ [s] | $z_c$ [m] |
|----------|------------|--------------------------|---------|-----|----------------|-----------|
| 55       | 10         | 2.5                      | 0.3     | 10  | 0.01           | 0.9       |

TABLE IV  
MPC PARAMETERS

| $\mathbf{v}_{max}^T$ ([m/s][m/s][deg/s]) | $\mathbf{a}_{max}^T$ ([m/s <sup>2</sup> ][m/s <sup>2</sup> ][deg/s <sup>2</sup> ]) | $\mathbf{j}_{max}^T$ ([m/s <sup>3</sup> ][m/s <sup>3</sup> ][deg/s <sup>3</sup> ]) |
|--|--|--|
| (1.5, 1.5, 180)                          | (3, 3, 360)  | (105, 105, 25200)  |

### A. Velocity MPC

Figures 3–4 show the results obtained with the velocity controller when including the jerk constraints. The robot follows the desired trajectory very well, quickly reducing the position error norm (to less than 1 cm in about 2 seconds) and then keeping it down to zero. After some oscillations, also the orientation error goes to zero. The input velocity commands and their first derivative stay always within the given bounds. In particular, the velocity is kept away from its limits and

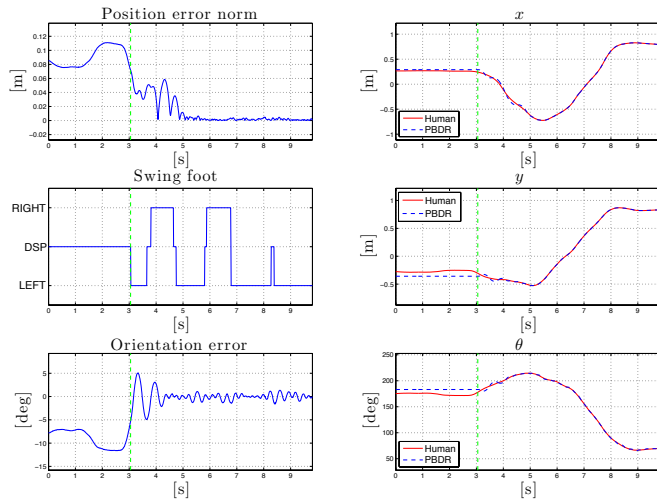


Fig. 3. Simulation results under velocity-level MPC. [Left] Norm of the position error, time diagram of the support phases, and orientation error. [Right] Position and orientation variables for the human and the robot.

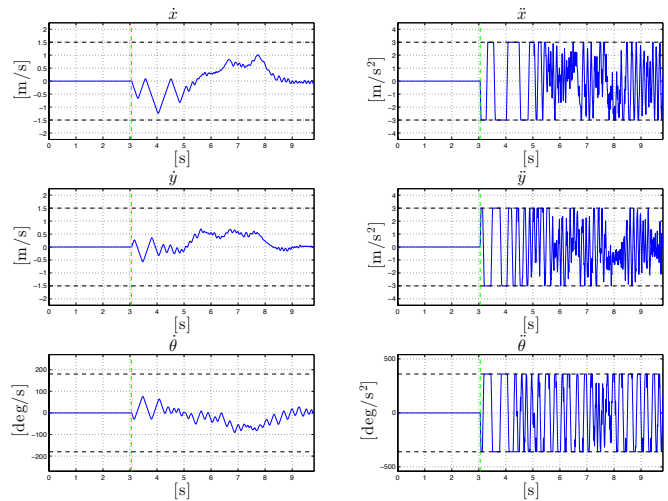


Fig. 4. Simulation results under velocity-level MPC. [Left] Velocity input commands. [Right] Input accelerations, as evaluated by numerical differentiation.

never saturates. Overall, the velocity control inputs have reasonably smooth profiles, albeit the accelerations undergo abrupt changes. In the absence of jerk bounds, this behavior was brought to an extreme situation, with the accelerations being almost always in saturation and bouncing between their upper and lower bounds also within a single time step.

### B. Acceleration MPC

For comparison, the same human trajectory was tracked using a MPC law designed at the acceleration level with jerk constraints. The results in Figs. 5–6 show that the evolution of the tracking errors are very similar to the previous case, although the velocity-level controller performs slightly better, especially in terms of position error. In the present control formulation, it is possible to provide an explanation of why the accelerations may change so abruptly and often saturate.

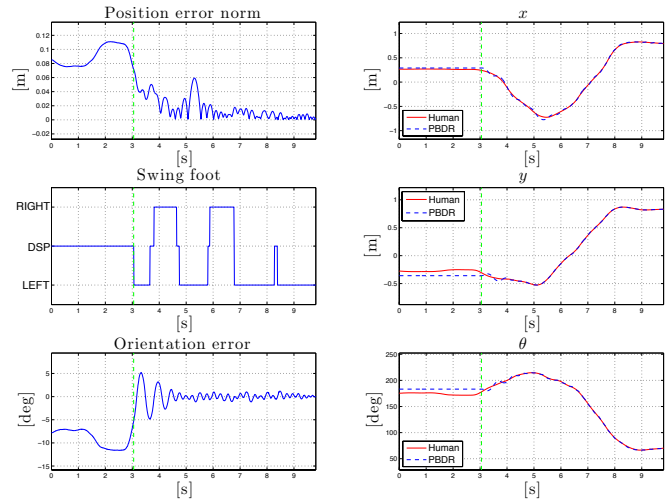


Fig. 5. Simulation results under acceleration-level MPC. [Left] Norm of the position error, time diagram of the support phases, and orientation error. [Right] Position and orientation variables for the human and the robot.

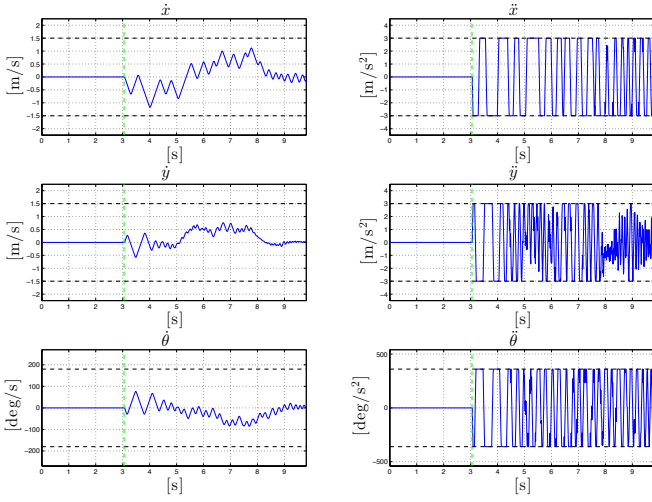


Fig. 6. Simulation results under acceleration-level MPC. [Left] Robot velocities. [Right] Acceleration input commands.

With reference to problem (24), the resulting matrix  $\tilde{W}$  becomes almost negligible when compared to  $\mathbf{h}$ , specifically is of order  $o(\Delta t^2)$ . This transforms the cost function, and thus the whole problem, into an almost linear one. In linear programming, solutions are always found on the vertices of the admissible region, and in the acceleration-level control solution these vertices correspond to saturated acceleration or jerk values. In the velocity-level control case, a similar mechanism may also come into play, but we found that the  $\tilde{W}$  matrix in (15) is no longer negligible.

## VI. EXPERIMENTAL RESULTS

Experiments were performed on the PBDR device available at Tohoku. Measurements for the human waist were obtained through ellipse-fitting from data provided by 2 LRFs installed on the robot, while 2 other LRFs provided measurements for the swing foot using the same method together with a clustering algorithm to distinguish between the two feet [12]. The measurement apparatus works at an output frequency of 28 ms. Thus, we have set  $\Delta t = 0.028$  s and  $N = 4$ , so that the preview window is kept to  $T = N \cdot \Delta t \approx 0.1$  s, just as in simulations. Moreover,  $d$  was set to 45 cm, in order to have a larger distance from the dancer. Implementation of the MPC controller was fast enough to run online, with no particular code optimization.

### A. Velocity MPC

For the velocity-level MPC, bounds on velocity, acceleration, and jerk have been set to:

$$\begin{aligned} \mathbf{v}_{max} &= (0.6 [m/s] \quad 0.6 [m/s] \quad 90 [deg/s])^T \\ \mathbf{a}_{max} &= (1.5 [m/s^2] \quad 1.5 [m/s^2] \quad 180 [deg/s^2])^T \\ \mathbf{j}_{max} &= (50 [m/s^3] \quad 50 [m/s^3] \quad 5730 [deg/s^3])^T. \end{aligned}$$

The results are reported in Figs. 7–8. The tracking error obtained in the experiments is slightly larger than in the simulations. This is mainly due to the tighter kinematic bounds imposed on the robot motion, which were chosen

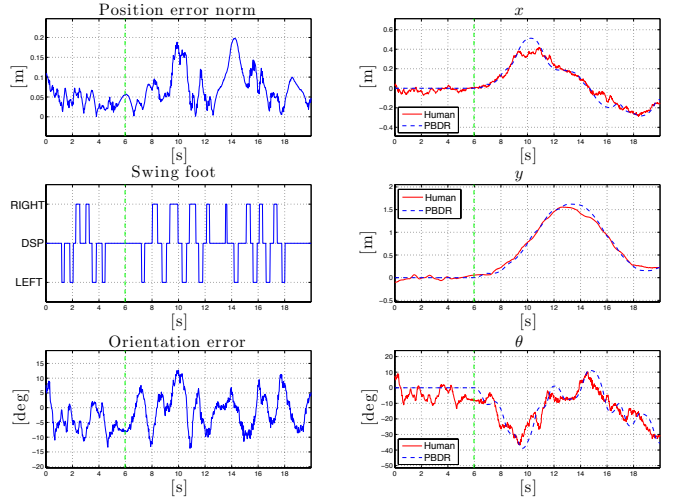


Fig. 7. Experimental results under velocity-level MPC. [Left] Norm of the position error, time diagram of the support phases, and orientation error. [Right] Position and orientation variables for the human and the robot.

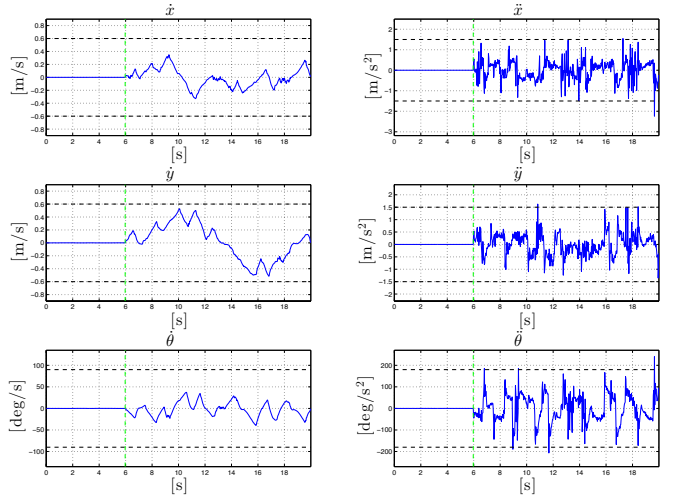


Fig. 8. Experimental results under velocity-level MPC. [Left] Velocity input commands. [Right] Acceleration inputs, as evaluated by numerical differentiation of the velocity commands.

for safety reasons. We also remark that the precision of the sensing system is lower when compared to the MoCap, and that the captured data are in general more noisy than those used for simulation. Nonetheless, the performance of the controller is still acceptable, with a maximum position error of 19.81 cm (with mean value 6.71 cm) and a maximum angular error of  $13.78^\circ$  (with mean value  $5.08^\circ$ ). Note that the acceleration occasionally exceeds the nominal bounds used for optimization (see Fig. 8). This can be explained by taking into account the imperfect following of the commanded velocity by the low-level servo loops. The accompanying video clip shows an experimental trial with this controller.

### B. Acceleration MPC

For the acceleration-level MPC, the commanded acceleration is converted by integration into a velocity reference input to low-level servo loops running at 1 KHz. A different set

## VII. CONCLUSIONS

A Model Predictive Control approach was used to let a ballroom partner robot follow naturally and accurately a human leader dancing waltz. Prediction of human motion during single-support phase was obtained using an Extended Kalman Filter applied to a simplified dynamic model consisting of a 3D Linear Inverted Pendulum and a low swinging foot. While intended for humanoid control, this model was found to provide reliable short-term predictions of the human behavior during waltz. Two different MPC controllers were designed and tested, solving on line an LQ problem with bounds on the robot differential kinematic variables and assuming velocity or acceleration as control input. The velocity-level MPC law was the most effective in experiments, producing fast and accurate reactive motion.

Among the possible extensions of this work, we plan to improve the design of the acceleration-level MPC law. In order to smooth further the command profiles, one could include a cost term in the objective function, proportional to the squared norm of the acceleration. Leader-follower interaction forces may also be taken into account, by introducing a human arm force model or, in a simpler form, via a virtual connection by a damped spring. The interaction forces could then be optimized within the MPC formulation.

## REFERENCES

- [1] H. Wang and K. Kosuge, "An inverted pendulum model for reproducing human's body dynamics in waltz and its application in a dance partner robot," in *Proc. IEEE/SICE Int. Symp. on System Integration*, 2010, p. 182–187.
- [2] —, "Towards an understanding of dancers' coupled body dynamics for waltz," in *Proc. IEEE/RJS Int. Conf. on Intelligent Robots and Systems*, 2011, p. 2008–2013.
- [3] —, "Understanding and reproducing waltz dancers' body dynamics in physical human-robot interaction," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2012, p. 3134–3140.
- [4] S. Kajita and K. Tani, "Study of dynamic biped locomotion on rugged terrain – Derivation and application of the Linear Inverted Pendulum Mode," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 1991, pp. 1405–1411.
- [5] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa, "The 3D Linear Inverted Pendulum Mode: A simple modeling for a biped walking pattern generation," in *Proc. IEEE/RJS Int. Conf. on Intelligent Robots and Systems*, 2001, p. 239–246.
- [6] J. Kim and J. Park, "Quick change of walking direction of biped robot with foot slip in single-support phase," in *Proc. 11th IEEE-RAS Int. Conf. on Humanoid Robots*, 2011, p. 339–344.
- [7] C. Bruni, G. D. Pillo, and G. Koch, "Bilinear systems: An appealing class of "nearly linear" systems in theory and applications," *IEEE Trans. on Automatic Control*, vol. 19, no. 4, pp. 334–348, 1974.
- [8] C. García, D. Prete, and M. Morari, "Model predictive control: Theory and practice – A survey," *Automatica*, vol. 25, no. 3, pp. 335–348, 1989.
- [9] A. Bemporad and M. Morari, "Robust model predictive control: A survey," in *Robustness in Identification and Control*, ser. Lecture Notes in Control and Information Sciences, A. Garulli, A. Tesi, and A. Vicino, Eds. Springer-Verlag, 1999, vol. 245, pp. 207–226.
- [10] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.
- [11] R. Byrd, M. E. Hribar, and J. Nocedal, "An interior point algorithm for large scale nonlinear programming," *SIAM J. on Optimization*, vol. 9, no. 4, pp. 877–900, 1999.
- [12] H. Wang, "Study of Coupled Dynamics in Physical Human-Robot Interaction towards a Cooperative Dance Partner Robot," Ph.D. dissertation, Dept. of Bioengineering and Robotics, Tohoku University, July 2012.

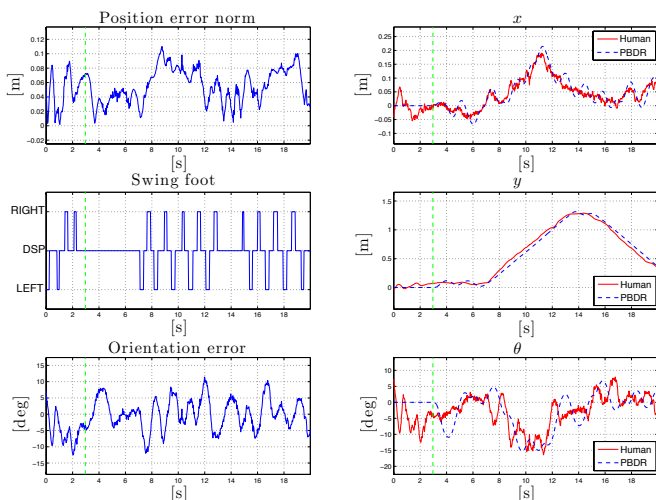


Fig. 9. Experimental results under acceleration-level MPC. [Left] Norm of the position error, time diagram of the support phases, and orientation error. [Right] Position and orientation variables for the human and the robot.

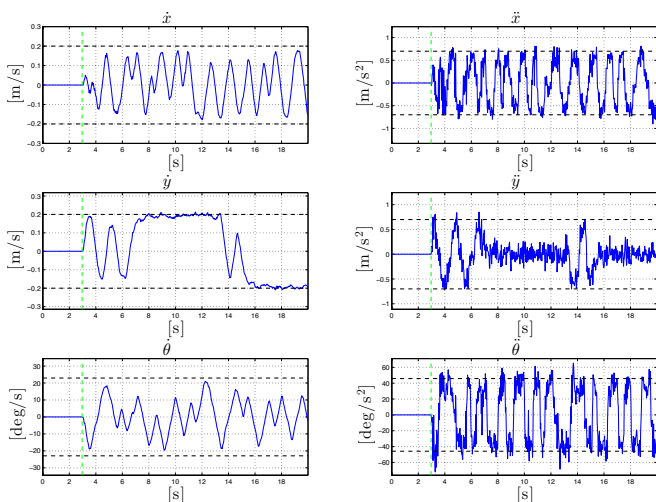


Fig. 10. Experimental results under acceleration-level MPC. [Left] Velocity inputs, as obtained by numerical integration of the acceleration commands. [Right] Computed acceleration commands.

of kinematic bounds has been imposed, indeed more strict than those used in the velocity-level control experiment:

$$\begin{aligned} \mathbf{v}_{max} &= (0.2 [m/s] \quad 0.2 [m/s] \quad 23 [deg/s])^T \\ \mathbf{a}_{max} &= (0.7 [m/s^2] \quad 0.7 [m/s^2] \quad 46 [deg/s^2])^T \\ \mathbf{j}_{max} &= (35 [m/s^3] \quad 35 [m/s^3] \quad 2005 [deg/s^3])^T. \end{aligned}$$

The results are reported in Figs. 9–10. It is apparent that the human is now moving quite slowly, in order to enable good tracking by the PBDR. The choice of restrictive bounds was rather a necessity, since the robot under acceleration-level MPC control was found to be less stable in the experiments (as opposed to simulations). This may be due to the excessive amount of kinetic energy stored by the robot under acceleration control. Moreover, the controller operates at a higher level of derivation and is thus less reactive; commanding a system via acceleration inputs usually requires a much higher control frequency than when using velocity inputs.