# Humanoid Dance Simulation Using Choreonoid and Whole-Body Model Predictive Control

Yuichi Tazaki†

***Abstract*— This workshop paper reports recent progress towards realizing dynamic dancing motion of humanoid robots by whole-body model predictive control. A reference key-frame sequence edited using Choreonoid is input a novel whole-body MPC algorithm based on the centroidal-dynamics and full-kinematics formulation. It is demonstrated in simulations that the model-predictive controller is capable of compensate for large angular momentum generated by rapid arm swinging motion.**

## I. INTRODUCTION

Humanoid robots are build to imitate the appearance and physical ability of human being. This characteristic is not only useful for replacing human labor but also for replicating the artistic skills of human experts with robots. Dance is an example of such artistic skills that involve dynamic and large movement of the entire body. The recent advances in the performance of robotic hardware paved a way for realizing acrobatic dances with humanoid robots. Establishing a design methodology of controllers that can realize stable dancing motion on a humanoid robot, even in simulation, is still a challenging and interesting topic to explore. Various existing studies focused on motion retargeting, which is to convert reference motion provided by motion capture or by manually edited choreography to dynamically feasible motion executable by a robot [1, 2]. Most existing retargeting methods, however, are based on simple balance control techniques that aims at stabilizing the center-of-mass (CoM), and the angular momentum dynamics is often neglected.

Recently, whole-body model predictive control has been actively studied and it has been used mainly to realize rough-terrain locomotion of biped and quadruped robots. To the author's knowledge, however, its application to dancing has not been reported in the literature. This workshop paper reports an initial progress of our attempt to apply whole-body MPC to humanoid dancing. As a first step towards realizing acrobatic dancing, we focus on the ability of whole-body MPC to compensate for large angular momentum generated by rapid arm swinging.

## II. CHOREOGRAPHY INTERFACE

Fig. 1 shows a screenshot of the GUI of Choreonoid [3] for dance motion editing. The GUI of Chorenoid is highly customizable. The one shown here is an example layout specialized for choreography, and consists of three main views: the scene view, the media view, and the pose-roll view. An orignal material of dance motion (e.g., CG animation, a video clip of human dance) is displayed in the media view, and the user edits key poses of the humanoid robot interactively in the scene view. The pose-roll view enables insertion and deletion of key frames as well as adjustment of timing of existing key frames. The edited choreography data can be exported in the YAML format to be loaded by the control program.

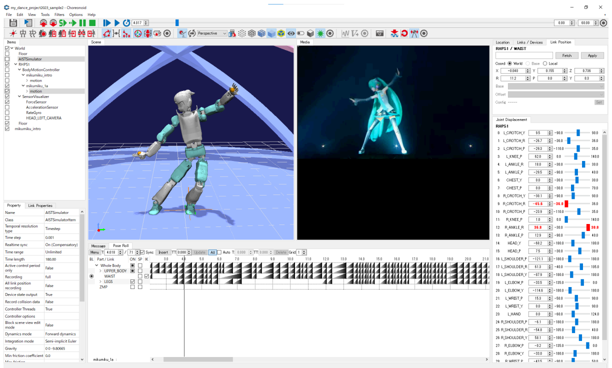

Fig. 1. GUI of Choreonoid

## III. WHOLE-BODY MODEL PREDICTIVE CONTROL

### A. Centroidal dynamics expressed in the CoM frame

The following gives a brief overview of our centroidal-dynamics/full-kinematics formulation. First, the movement of the CoM is expressed as

$$\dot{\boldsymbol{p}}_{\mathrm{com}} = \boldsymbol{v}_{\mathrm{com}}, \quad \dot{\boldsymbol{v}}_{\mathrm{com}} = \frac{1}{m} \sum_{i \in 1:n} \boldsymbol{f}_i$$

where $m$ is the total mass, $\boldsymbol{f}_i$ is the external force applied to the $i$-th contact, and $n$ is the number of contacts. To express the angular momentum dynamics, we introduce a special coordinate frame whose origin is attached to the CoM of the system and its orientation is synchronized with that of the base link. The total angular momentum around the CoM is expressed as

$$\boldsymbol{L} = I \boldsymbol{\omega}_{\mathrm{f}} + \boldsymbol{q}_{\mathrm{f}} \hat{\boldsymbol{L}}.$$

Here, $\boldsymbol{q}_{\mathrm{f}}$ and $\boldsymbol{\omega}_{\mathrm{f}}$ are the orientation and angular velocity of the base link, respectively, and $I$ is the instantaneous inertia matrix of the whole system. Moreover, $\hat{\boldsymbol{L}}$ is the *local* angular momentum of the system; namely, it expresses the total angular momentum of the movement of the links relative to the CoM. The above equation is essentially the same

---

[†] The author is with Faculty of Engineering, Department of Mechanical Engineering, University of Kobe, 1-1 Rokkodai, Nada-ku, Kobe, Japan. `tazaki@mech.kobe-u.ac.jp`

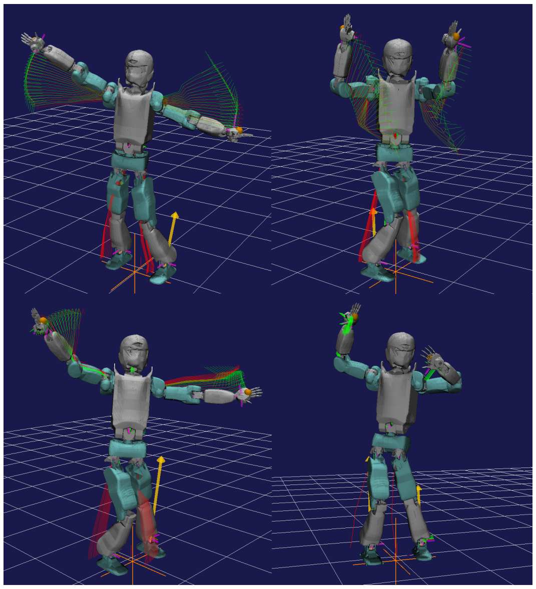

Fig. 2. Screenshots from dancing simulation

as the centroidal angular momentum equation. In [4], this relationship was directly used to express the time derivative of the base link orientation with the angular momentum. In our approach, on the other hand, we differentiate it one again to obtain

$$\dot{\boldsymbol{\omega}}_{\mathrm{f}} = I^{-1}\Big( -(\dot{I}\boldsymbol{\omega}_{\mathrm{f}} + \boldsymbol{\omega}_{\mathrm{f}} \times (\boldsymbol{q}_{\mathrm{f}}\hat{\boldsymbol{L}}) + \boldsymbol{q}_{\mathrm{f}}\dot{\hat{\boldsymbol{L}}}) + \sum_{i \in 1:n} (\boldsymbol{\tau}_i + (\boldsymbol{p}_i - \boldsymbol{p}_{\mathrm{com}}) \times \boldsymbol{f}_i)\Big)$$

where $\boldsymbol{\tau}_i$ is the external moment applied to the $i$-th contact. The state and input variables are defined as follows.

$$\boldsymbol{x} = \left[\boldsymbol{p}_{\mathrm{com}}; \boldsymbol{v}_{\mathrm{com}}; \boldsymbol{q}_{\mathrm{f}}; \boldsymbol{\omega}_{\mathrm{f}}; \boldsymbol{\theta}; \dot{\boldsymbol{\theta}}\right],$$
$$\boldsymbol{u} = \left[\ddot{\boldsymbol{\theta}}; \boldsymbol{f}_1; \ldots; \boldsymbol{f}_n; \boldsymbol{\tau}_1; \ldots; \boldsymbol{\tau}_n\right]$$

Here, $\boldsymbol{\theta}$ is the joint angle vector. The centroidal equations together with the update law of joint angles are brought together and discretized in time to obtain a discrete-time state equation $\boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, \boldsymbol{u}_k)$. An optimal control problem (OCP) is formulated as a minimization of a cost function subject to a set of constraints. The cost function consists of terms related to desired feet pose and desired joint angles of the upper body joints. The desired values are given by interpolating the key poses of dance motion. Constraints consist of joint angle limits and feasibility conditions of contact wrenches. Custom C++ code implementing differential dynamic programming (DDP) allowing infeasible starting points is used for solving the OCP.

## IV. DANCE SIMULATION RESULTS

A model of RHP-S1 "Friends" (Kawasaki Heavy Industry) was used in simulation. The model has 32 DoF (4 DoF in torso and neck, 8 DoF in each arm, 6 DoF in each leg). The kinematic and inertial parameters were matched to the real robot. The total mass of the robot is about 50kg. Choreonoid was used for both dance motion editing and dynamical simulation. The choreography is based on the introductory part of the dance performance of a virtual character Hatsune Miku [5]. The edited choreography data consisted of 26 keyframes with a total duration of 11s. The AIST simulator item was used for physics simulation. The time step of physics simulation was set as 0.25ms, whereas the control cycle was set as 1ms. The prediction horizon of MPC was set as 200ms which was split into $N = 10$ discrete time step with a uniform step size of 20ms. The the update cycle of MPC was set as 5ms. To work around the difficulty of state estimation, full state information of the robot are directly obtained from the simulator.

Some screenshot images of simulation are shown in Fig. 2. See the Youtube video [6] for better visualization of simulation. The knee joints of the robot have to be almost fully streched throughout the tested dance motion. This makes it difficult to apply conventional balance compensation techniques based on CoM shifting. Instead, the model-predictive controller makes use of tilting of the upper body and adjustment of arm trajectory to maintain the centroidal state inside the stabilizable region. In fact, when the single rigid body model was used instead of the proposed CDFK model, the controller could not compensate for large reaction force of arm swinging, which resulted in large tracking error or falling in the worst case.

## V. OUTLOOK

In the workshop, we will show more acrobatic dance simulation results including stepping and jumping.

## REFERENCES

[1] K. Yamane and J. Hodgins. Simultaneous tracking and balancing of humanoid robots for imitating human motion capture data. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2510–2517, 2009.

[2] S. Nakaoka, A. Nakazawa, F. Kanehiro, K. Kaneko, M. Morisawa, and K. Ikeuchi. Task model of lower body motion for a biped humanoid robot to imitate human dances. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3157–3162, 2005.

[3] S. Nakaoka. Choreonoid: extensible virtual robot environment built on an integrated gui framework. In *2012 IEEE/SICE International Symposium on System Integration (SII)*, pages 79–85, 2012.

[4] J. Sleiman, F. Farshidian, M. Minniti, and M. Hutter. A unified mpc framework for whole-body dynamic locomotion and manipulation. *IEEE Robotics and Automation Letters*, 6(3):4688–4695, 2021.

[5] Hatsune miku magical mirai 2017. `https://www.youtube.com/watch?v=RU-OAZas1Ps`.

[6] Humanoid dance simulation. `https://www.youtube.com/watch?v=6gC6EcsDwo0`.

**Yuichi Tazaki** He received the Dr.Eng degree in control engineering from Tokyo Institute of Technology, Tokyo, Japan, in 2008. From 2007 to 2009, he was a research fellow of Japan Society for the Promotion of Science. In 2008, he was a guest scientist of Honda Research Institute Europe, Germany. From 2009 to 2016, he was an assistant professor of Nagoya University, Japan. From 2016, he has been an associate professor of Kobe University, Japan. His research interests include autonomous vehicles, mobile robots, and humanoid robots. He is a member of IEEE, SICE and The Robotics Society of Japan.