Waltz right turn with a Humanoid Robot using Inverse Kinematics

MATHÉMATIQUES VISION APPRENTISSAGE

Constantin Vaillant-Tenzer [1 3] Charles Monte [2 3]

[1]Ecole Normale Supérieure - PSL [2]Telecom Paris [3]Master MVA

Introduction

Our aim is to code a robot able to dance a waltz right turn using **inverse kinematics**.

Previous works

Waltz dancing is not a common research topic among dancing robots. However, to help robots learn how to dance, there are some already known methods like **learning from observation**(Nakaoka et al. 2007) where the robot chooses from a series of leg tasks the good ones to imitate a dancing human; or **real-time gesture responsive frameworks**(Hong et al. 2024) where the robot generates unpredictable artistic responses to a dancer's movements.
Concerning waltz dancing, a **dance partner robot**(Kosuge et al. 2011) has been developed to realize human-robot coordination by estimating the next intended movement of a human through physical interaction.

The waltz right turn movement in theory

The movement made during a right turn of the waltz dance can be well described using the movement of the center of gravity of the person dancing.
During the dance, the person's movement is split in two summed parts: while moving along the perimeter of an ellipse - most commonly referenced as **the ball circle** - the dancer turns on himself along the perimeter of a circle with a diameter close to 1 meter. Both those translations can be estimated to be of constant angular velocities.
The dancer does a full turn around the circle every 360 beats of the music (of known BPM) so the angular velocity around the circle is $\omega_C = \frac{-2\pi \text{BPM}}{6*60}$ while we estimate the dancer to go around the ball room every minute, meaning the angular velocity around the ellipse is $\omega_E = \frac{-2\pi}{60}$.
The movement of the center of gravity we aim at reproducing looks like this from above. It has been simulated for 2 minutes with a 187 BPM music.

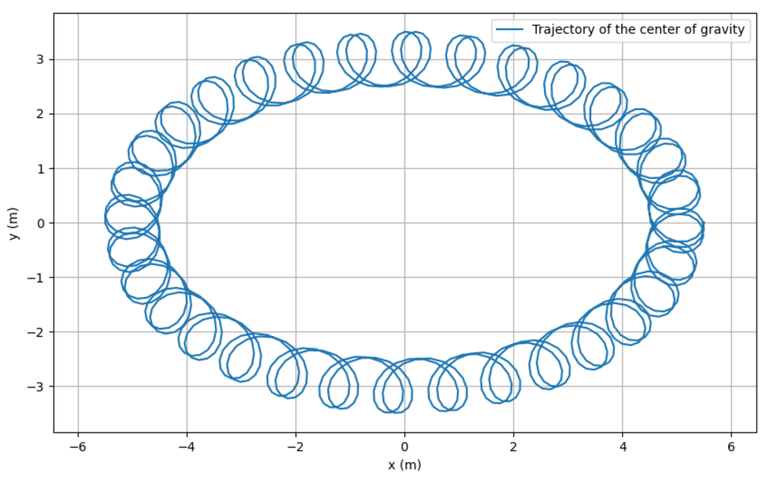

Figure 1. Simulated movement of a Waltz dancer's center of gravity for 2 minutes

Our approach of the problem

To simulate the trajectory done by a dancer with a robot, we have trained a humanoid robot to follow the sequence of movements done by the dancer during a waltz right turn.
To do so, we first had to make the assumption that gravity and other forces applied to the dancer does not affect their movement. We also noticed that robots do not have to produce an effort to keep their chest upright, which means a robot will lead the movement with its feet, while humans do the opposite. Thus, the rotations will exclusively be produced thanks to the lower half of the robot body.
We then had to acquire the kinematics of the waltz right turn with the help of a video and then solve inverse kinematics with a humanoid robot model to fit the acquired movement. After having a robot able to dance the waltz right turn, we only had to fit the robot's movement to the music BPM, by making sure a right turn had the right duration.

Our approach's pipeline

We did not have sufficient time to implement everything ourselves from kinematics acquisition to inverse kinematics solving so we decided to center ourselves around acquiring a good base for the movement kinematics. Thus, we mainly relied on already implemented blocks to fit inside our pipeline while trying to get the best approximation of a waltz right turn possible.
The global pipeline we followed for our implementation is the following.

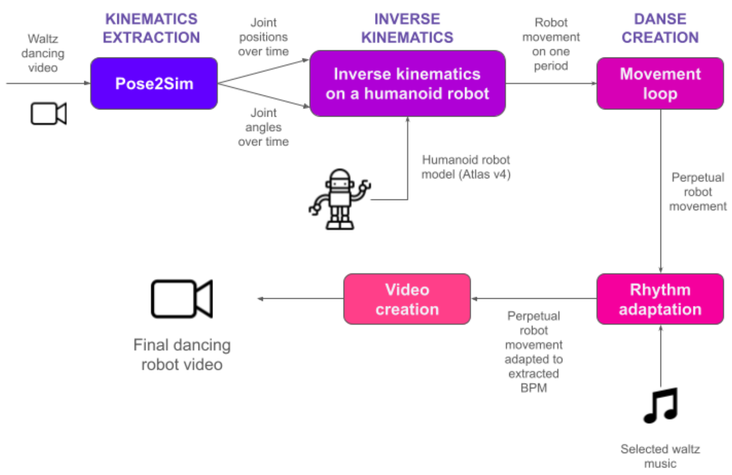

Figure 2. Our approach's full pipeline

Already implemented blocks used

Kinematics acquisition

After unsuccessful attempts to extract from proposed (Z. Li et al. 2019), or discovered (J. Li et al. 2023) solutions both the joint positions and angles, we decided to use a pipeline that can extract both, **Pose2Sim** (Pagnon, Domalain, and Reveret 2021).

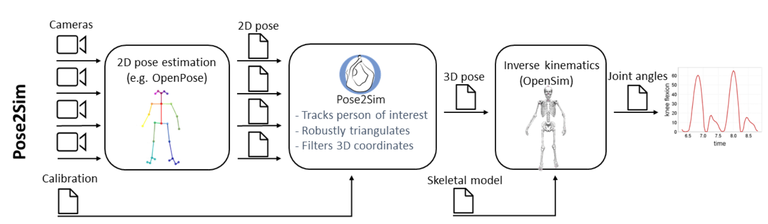

Figure 3. The Pose2Sim pipeline

This solution extracts 2D keypoints coordinates (using RTM Pose by Jiang et al. 2023) to produce an OpenSim result (full-body 3D joint angles), which we use in our pipeline by extracting the information that is interesting from us, ie. the RTMPose output and the full-body 3D joint angles.

Inverse Kinematics solving

To solve the inverse kinematics, we applied Pink (Caron, De Mont-Marin, et al. 2024) which solves differential inverse kinematics by weighted tasks.
The method uses **residual functions** of the robot configuration q that should be driven to zero to make the robot do a certain task. In our case, we for instance try to put the robot's feet at a certain position p^*_{feet} so an example of a residual would be $e(q) = p^*_{\text{feet}} - p_{\text{feet}}(q)$.
To solve the equation system produced, the method computes a velocity v that satisfies the equation $J_e(q)v = \dot{e}(q) = -\alpha e(q)$ - $J_e(q)$ being the Jacobian of task e - for each residual. It is of course not possible so the method finds the optimal solution to the following minimization problem, which finds the movement which tries to solve all of the tasks at the same time.

$$\min_v \sum_{\text{tasks } e} ||J_e(q)v + \alpha e(q)||^2$$
$$\text{subject to} \quad v_{min}(q) \leq v \leq v_{max}(q)$$

Choice of a humanoid robot model

We explored the Robot descriptions in Python repo (Caron, Romualdi, et al. 2024) to find a humanoid robot with similar joints to our Pose2Sim output, we finally decided to use AtlasV4 as our humanoid model.

Our results

We got a robot able to dance a waltz right turn with a good approximation of the movement done by a human. The robot is able to follow the movement of the center of gravity of a human dancing a waltz right turn. You can seea video of the robot dancing and the GitHub repository of the project by scanning the QR code:

Next steps

While the results already are satisfying, we identified two further steps that could be added to the pipeline to achieve even better results :

1. Adding a controlling module to maintain a constant angular velocity on the small circle. This cannot be done by humans and will make the movement look even better.
2. Take care of the robot's balance throughout the movement ;
3. Be able to dance the waltz right turn with a partner.

References

Caron, Stéphane, Yann De Mont-Marin, et al. (2024). *Pink: Python inverse kinematics based on Pinocchio*. Version 3.1.0. URL: `https://github.com/stephane-caron/pink`.

Caron, Stéphane, Giulio Romualdi, et al. (2024). *robot-descriptions.py: Robot descriptions in Python*. Version 1.13.0. URL: `https://github.com/robot-descriptions/robot_descriptions.py`.

Hong, Hui-Ting et al. (2024). "A Dance Performance with a Humanoid Robot using a Real-time Gesture Responsive Framework". In: *2024 33rd IEEE International Conference on Robot and Human Interactive Communication (ROMAN)*, pp. 1550–1555.

Jiang, Tao et al. (2023). *RTMPose: Real-Time Multi-Person Pose Estimation based on MMPose*.

Kosuge, Kazuhiro et al. (Jan. 2011). "Partner Ballroom Dance Robot -PBDR-". In: *SICE Journal of Control, Measurement, and System Integration* 1.1, pp. 74–80.

Li, Jiefeng et al. (June 2023). "NIKI: Neural Inverse Kinematics with Invertible Neural Networks for 3D Human Pose and Shape Estimation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Li, Zongmian et al. (June 2019). "Estimating 3D Motion and Forces of Person-Object Interactions From Monocular Video". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp. 8632–8641.

Nakaoka, Shinichiro et al. (Aug. 2007). "Learning from Observation Paradigm: Leg Task Models for Enabling a Biped Humanoid Robot to Imitate Human Dances". In: *I. J. Robotic Res.* 26, pp. 829–844.

Pagnon, David, Mathieu Domalain, and Lionel Reveret (2021). "Pose2Sim: An End-to-End Workflow for 3D Markerless Sports Kinematics—Part 1: Robustness". In: *Sensors*.

constantin.tenzer@ens.psl.eu charles.monte@telecom-paris.fr