Boxing Robot: Design and Motion Tracking

Wenxi Chen
Mechanical Engineering
Columbia Univeristy
wc2746@columbia.edu

Chenfei Zhu
Mechanical Engineering
Columbia Univeristy
cz2633@columbia.edu

Wenjie Lin
Mechanical Engineering
Columbia University
wl2789@columbia.edu

Yinan Wang
Mechanical Engineering
Columbia Univeristy
yw3692@columbia.edu

Chenxi Tao
Mechanical Engineering
Columbia University
ct3007@columbia.edu

***Abstract*—In movies, it's so cool to watch the boxing robots fighting with each other imitating human's actions. This paper aims to make it possible in real life. We present a boxing robot to mimic human's movements using only visual input. With optical cameras without markers, deep learning model and post-processing algorithms, 3d pose reconstruction has been made successfully. To be precise, PD control is implemented as well.**

Keywords—boxing robot, motion capturing, 3d pose reconstruction, robot dynamics, PD control

I. Introduction

Motion capturing is an active area of research. F. Siles et al. [1] came up with an optical motion capture system with markers and not in real time. Yongbin Li et al. [2] focused on the improvement of the speed of pose construction using fuzzy control. Seul Jung et al. [3] used an exoskeleton type motion capturing device to collect motion data. A lightweight convolutional neural network architecture for human pose estimation called BlazePose [4] was presented. Jonas Koenemann et al. [5] used a compact human model and considered the positions of the end effectors as well as the center of mass to realize a statically stable pose construction. In [6], kinematic and dynamical mappings for human-likeness and stability were utilized for motion capturing for humanoid robots.

We desire to design a boxing robot using visual input to enhance mobility. Furthermore, achieving 3d pose reconstruction based on markerless optical cameras is a goal as well.

For our uniqueness, we only use optimal cameras rather than optical motion capture systems or infrared cameras, accompanied with a deep learning model and some post-processing algorithms to reconstruct the 3D motion data.

II. System Design

A. Robot arm design in simulator

Constructed simulator is using Rigid Body Tree structure in MATLAB robotics toolbox. All links are connected by a single revolute joint. Robot base is located at the waist of the robot. First link is the trunk connecting two shoulders and is able to rotate horizontally, twisting two shoulders forward and backward. Link 2, 3, 4 endowed the upper arm with 3 degrees of freedom, in which joint 2,3 locates the elbow point. While joint 4 decides the rotation of the upper arm around the axis of the upper arm itself thus determines the location of wrist point along with joint 5. The other arm has symmetry set up with link and joint 6, 7, 8, 9 based at link 1. The length parameter of this robot is defined as follows: Shoulder height to base is 0.3m, shoulder width is 0.6m, the length of upper arm is 0.4m, the length of lower arm is 0.3m.

The dynamic properties of robot arms are set as follows: trunk has mass of 8kg, each upper arm and lower arm has mass of 6kg, 2 links on the shoulder and the end effector wrist all have mass of 0.5kg each.

Simulator is built based on Forward Dynamics in MATLAB, simulation time step is set to be 0.01s, each time stamp the joints' angle, angle velocity and angle acceleration are updated based on controller torque and dynamic properties.

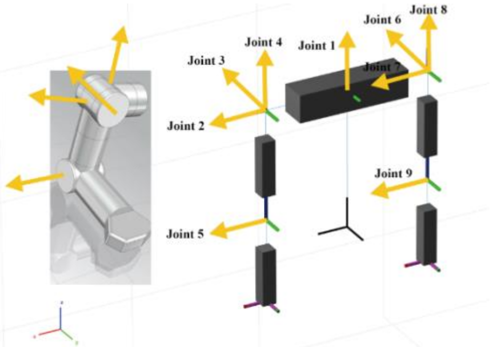

FIG. 1. Arrangement of robot in simulator. This robot has a body, 2 upper arms, 2 lower arms and 9 joints. Directions of joints are denoted by their rotation axes.

B. Visual data processing pipeline

1. Motion data collect

To collect motion data, we utilized MediaPipe Pose, a ML solution for high-fidelity body pose tracking, which infer 33 3D body landmarks on the whole body from RGB video frames, to track the motion of 8 upper body landmarks when our model is boxing.

In our experiment, we used two cameras placed on the front left and front right side of our model separately to avoid prediction error brought by occlusion. To ensure that motion data extracted from different videos are synchronized, we used sound signals to synchronize our videos with MATLAB. After the acquisition of videos, we analyzed the amplitude diagram of sound signals in both videos, and trimmed them from the second peak to synchronize them. Then, we fed these synchronized videos to MediaPipe Pose [7] model to get the motion data from both views.

2. 3D data reconstruction

After capturing 3D coordinate data in two video streams, we establish two coordinate frames with center based on the center of tracked two shoulder points. z direction is defined as upwards in the video frame, x direction is towards the right side in video, and y is perpendicular to the plane of each video frame and pointing inside the planes of the video frame. In the result of mediapipe, x and z coordinates correspond to location of body parts in each frame, y coordinates are given by the depth information inferred by the MediaPipe algorithm [7].

To align the two coordinates information into one, firstly we move one of the coordinate frames by the vector $\hat{d}$ connecting two frame bases. Then we need to calculate the rotation matrix between these two frames, in our example case two cameras are filming at roughly the same height and z coordinates both upwards, the only rotation we need is around the z axis. By calculating the angle between vectors from one shoulder to another in two frames, we have the rotation angle around the z axis. So, the homogeneous transformation matrix from frame 2 to frame 1 is given as:

$$H_1^2 = \begin{bmatrix} R_{z,\emptyset,3\times3} & \hat{d}_{3\times1} \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix} \tag{1}$$

Since boxing sometimes cause one arm invisible from one of the cameras and thus leads to the recognition of this arm being unreliable. So we abandon the far side arm data in each camera view and only use the visible side body parts including shoulder point, elbow point and wrist point. In this way we combine the two camera data into body parts tracking results in a coordinate system.

Lastly, to reduce the trembling noise of recognition results which could lead to explosive joint velocity, we apply a moving average on near 10 frames to filter the coordinate signal.

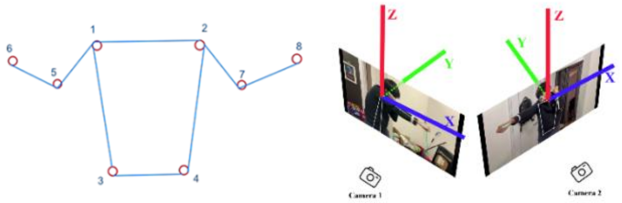

FIG. 2. A: 8 body landmarks tracked: 1-right shoulder; 2-left shoulder; 3-right hip; 4-left hip; 5-right elbow; 6-right wrist; 7-left elbow; 8-left wrist; B: The arrangement of two optical cameras.

C. *Robot arm controller*

1. Solving inverse kinematic

After obtaining the desired trajectory of each joint point from human movement, we need to solve the inverse kinematics problem here to get the joint angle for the corresponding pose in each frame. In this case, we are having all the joint locations and there is a clear only solution for the problem. By solving simple geometry of angle between two vectors, we can get all the joint angles.

2. PD controller for trajectory tracking

Firstly we calculate the target velocity of each joint by using a second order numerical differentiation. Then, or a dynamic trajectory tracking problem, we adapted PD controller, the algorithm is illustrated as follow:

$$Torque = K_p \cdot (q_{desire} - q_{actual}) + K_d(\dot{q}_{desire} - \dot{q}_{actual}) \tag{2}$$

K_p and K_d are proportional and differential controller coefficients, they are separately assigned to 9 joints, though manually tuning these parameters we tried to optimize the controller effect under the limit of motor torque. The twisting of trunk is rotating the whole body with large rotational inertia, so we adjust the output limit of joint 1 motor as 300 Nm. The joints on the shoulders are set to be capable of enforcing 200 Nm, the elbow joint is set to have limit of 100 Nm. Under these limitations, the robot arms are capable of reproducing most of the human arm motion, controller performance is presented in the next section.

D. *Hardware design*

1. Arm parameter

TABLE. 1. Key Parameters Of Robot Arms

Part	Length (mm)	Diameter (mm)	Weight (g)	Material
Upper arm	311.976	120	1080.346	PP
Lower arm	265	120	1189.414	PP

2. Joint motor parameters

TABLE. 2. Key Parameters Of Joint Motors.

Part	Type	Voltage (v)	Power (w)	Rated Torque (nm)	Rated Speed (rpm)
Elbow	LSG-32-142-xx	24-48	1200	99	58.8
Shoulder	LSG-32-142-xx	24-48	1200	153	37.0

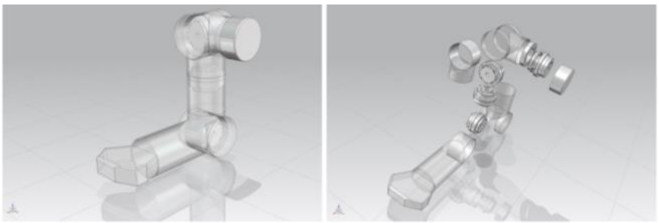

FIG. 3. Assemblies and exploded diagrams of one robot arm.

III. System Analysis

A. *Result for Video Synchronization (based on audio)*

Use audio signals to synchronize two videos, the max error is about 30ms (one frame).

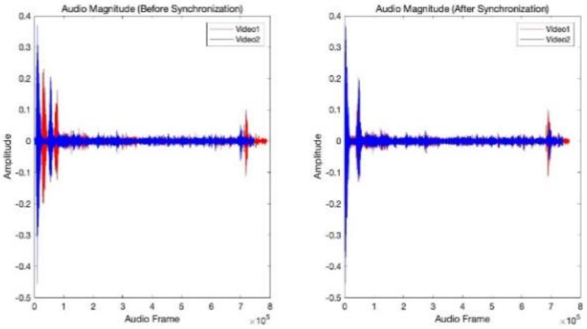

FIG. 4. Amplitude diagrams of sound signals in two videos by frames. Left: before synchronization; Right: after synchronization.

B. *3D data workspace projection*

We analyze the workspace of designed 9-DOF robot arms and project the processed data to executable workspace. Length between tracked joints are fluctuating in each frame, we maintain the orientation of shoulder, upper arms and lower arms, and use these directions to project the joint location from the previous base link and with proper length between them. In this method, we have gained the target trajectory of each robot joint within our robot's workspace.

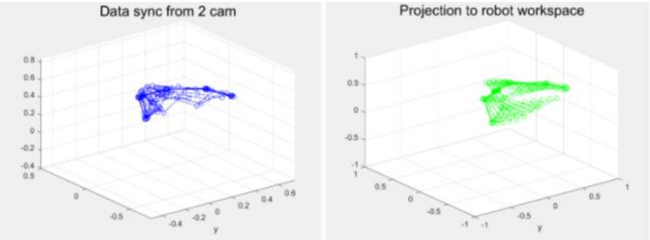

FIG. 5. 3D trajectory of body landmarks in one punch.

Left: before projection; Right: after workspace projection.

C. PD controller response analysis

Figure 6 shows several frames of output robot configurations and target configurations, we can tell the overall pose and movement is reproduced by the controller, and dynamic example clips can be viewed in the link mentioned below.

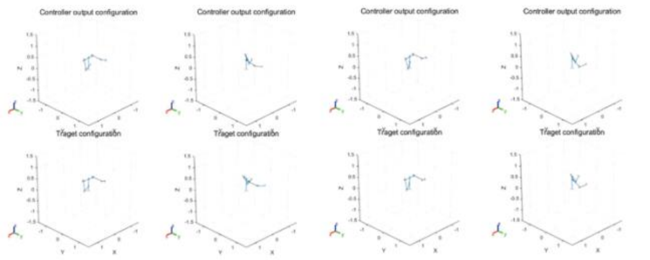

FIG. 6. Controller response (first row) compared to target configuration (second row) [Link to the robot video]

A clip of joint response compared to the target trajectory is shown in the figure 7, the robot arm can reproduce desired movement generally though the angle may not be exactly the same, which is acceptable considering the target trajectory may contain unreasonable fluctuations caused by the data processing. Trunk joint torque frequently meet the output limit of 300 N·m which corresponds to the fact that boxing is a sport requiring core strength to fastly turn the whole body.

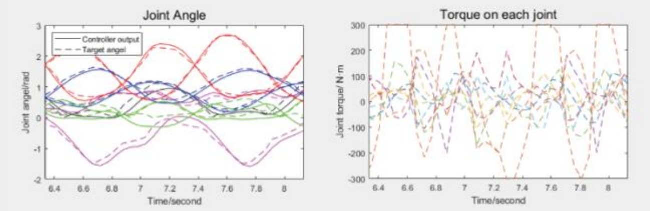

FIG. 7. Controller trajectory tracking performance: (a) joint angle response compared to target joint trajectory; (b) the output torque.

We calculated the average absolute response angle error of each joint, most of errors are below 10 degrees. Joint 7, joint 8 are on the shoulder of the left arm, their target trajectory contains extra more fluctuations which increase the difficulty for the controller.

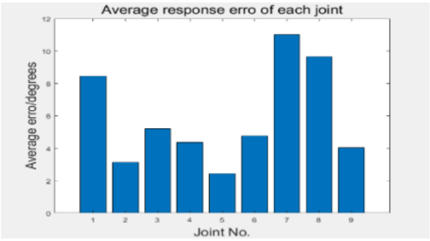

FIG. 8. Average error of controller output in each joint.

IV. Discussion

Based on the result of controller output, we can tell the quality of captured 3D pose would significantly influence the difficulty of reproducing the movement, noise of high frequency shaking could lead to extra burden on the controller. A more sophisticated algorithm and our hardware can also be improved like using professional cameras could be used to obtain more accurate pictures.

We noticed that to mimic fast human motion, we had to provide a large torque of nearly 200-3hundres. Also it requires the structure to have enough strength to bear with the self-generated force. To reproduce the flexibility and mobility of human movement, a better mechanical design must be achieved with less structure weight and motors that are capable of executing explosive torque to meet the speed of human boxing. Also, electric motors may not be able to meet the requirements here, a different power system like the electro-hydraulic system used on atlas might be an alternative choice [8].

V. CONCLUSION

In conclusion, we designed a structure and assembled a kind of humanoid robot arm with 9 degrees of freedom. We reconstructed the 3D position of human landmarks based on multi-cameras. For the trajectory following, we tuned the PID controller successfully under the torque limit. Finally, our robot arm moves in a way extremely like human boxing motion in real life.

Our next step is to integrate all the modules and make the real-time edition, which means we may need more advanced algorithms for reconstructing 3D data and meanwhile, a more suitable rationality of the mechanical structure.

VI. Reference

[1] Dajles, D., & Siles, F. (2018, July). Teleoperation of a Humanoid Robot Using an Optical Motion Capture System. In 2018 IEEE International Work Conference on Bioinspired Intelligence (IWOBI) (pp. 1-8). IEEE.

[2] Li, Y., Wang, P., Fang, Y., & Wang, D. (2019, August). Design and implementation of a boxing robot based on fuzzy control. In Journal of Physics: Conference Series (Vol. 1303, No. 1, p. 012065). IOP Publishing.

[3] Jeon, P. W., & Jung, S. (2003, July). Teleoperated control of mobile robot using exoskeleton type motion capturing device through wireless communication. In Proceedings 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2003) (Vol. 2, pp. 1107-1112). IEEE.

[4] Bazarevsky, V., Grishchenko, I., Raveendran, K., Zhu, T., Zhang, F., & Grundmann, M. (2020). BlazePose: On-device Real-time Body Pose tracking. arXiv preprint arXiv:2006.10204.

[5] Koenemann, J., Burget, F., & Bennewitz, M. (2014, May). Real-time imitation of human whole-body motions by humanoids. In 2014 IEEE International Conference on Robotics and Automation (ICRA) (pp. 2806-2812). IEEE.

[6] Kim, S., Kim, C., You, B., & Oh, S. (2009, October). Stable whole-body motion generation for humanoid robots to imitate human motions. In 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 2518-2524). IEEE.

[7] Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., ... & Grundmann, M. (2019). Mediapipe: A framework for building perception pipelines. arXiv preprint arXiv:1906.08172.

[8] Kuindersma, S., Deits, R., Fallon, M. et al. Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot. Auton Robot 40, 429–455 (2016). https://doi.org/10.1007/s10514-015-9479-3