Proceedings of Machine Learning Research vol vvv:1–4, 2025

Supplemental Material
BIGE : Biomechanics-informed GenAI for Exercise Science

Shubh Maheshwari SHMAHESHWARI@UCSD.EDU
Anwesh Mohanty ANMOHANTY@UCSD.EDU
Yadi Cao YAC066@UCSD.EDU
Swithin Razu SRAZU@UCSD.EDU
Andrew McCulloch AMCCULLOCH@UCSD.EDU
Rose Yu ROSEYU@UCSD.EDU
University of California, San Diego
La Jolla, CA 92093

Contents

1. Temporal Segmentation

In each sample, an athlete performs an exercise $3 - 4$ times. Each action is segmented out to make the training data consistent.

For the temporal segmentation, that is to find the start and the end frame for each exercise iteration. We found out that the angular velocity of the joints is a good indicator. Fig. **??** in the image shows the angular velocity for each joint (in deg/s) for each timestep. The red peaks in the bottom plot showcase the timestep at which the activity ends.

For squats, we use the maximum of the left and right knee flexion to identify the start and stop points using peaks or dips. I typically take the earliest and latest time points from the combined data of both knees. While this task can be time-consuming, it's essential for accuracy.

You probably only need (a) pulling the start stop cycles and time normalizing them, there are two steps for this. As a result of this the peaks will be generally aligned but you don't want to force it since this variability is important. Clinically the peaks are important but we usually only care about the max or min and not the time stamp.

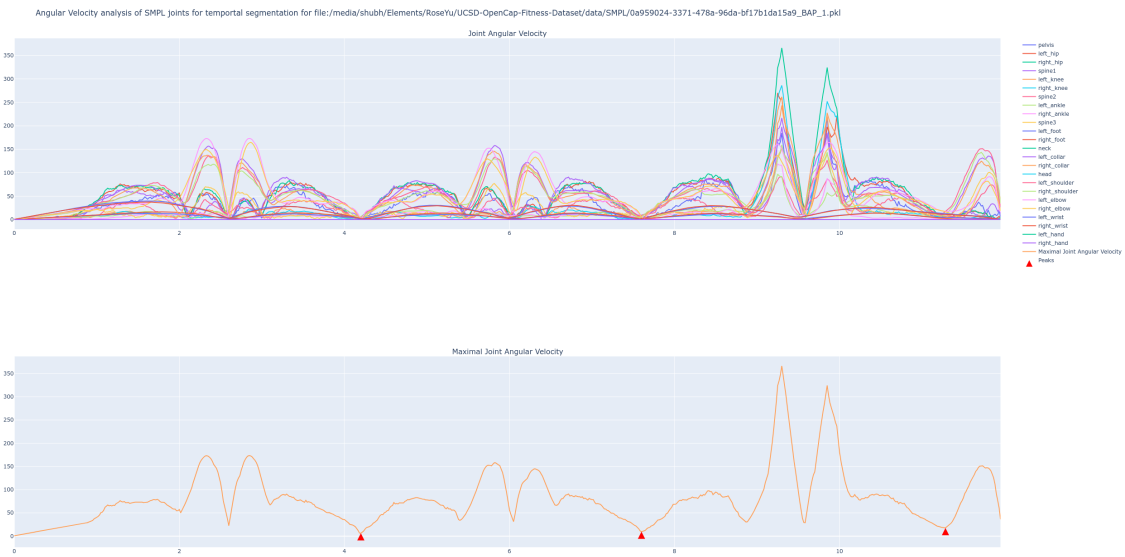

Figure 1: The top plot in the image shows the angular velocity for each joint (in deg/s) at each timestep. The red valley in the bottom plot indicates the timestep at which the activity ends.

2. Muscle Simulation Setup

Finally, OpenCap can estimate dynamics using muscle-driven tracking simulations of joint kinematics. The tracking simulations are formulated as optimal control problems that aim to identify muscle excitations that minimize a cost function subject to constraints describing muscle and skeletal dynamics. The cost function J (Equation 1) includes squared terms for muscle activations (a) and excitations of the ideal torque motors at the lumbar, shoulder, and elbow joints (e_{tm}). It also includes tracking terms (squared differences between simulated and reference data), namely tracking of experimental joint positions ($\boldsymbol{q}$), joint velocities ($\dot{\boldsymbol{q}}$), and joint accelerations ($\ddot{\boldsymbol{q}}$):

The musculoskeletal model is driven by 80 muscles actuating the lower-limb coordinates and 13 ideal torque motors actuating the lumbar, shoulder, and elbow coordinates. Ground reaction forces (i.e., external forces) are modeled through six foot-ground contact spheres attached to the foot segments of the model. Raasch's model is used to describe muscle excitation-activation coupling, and a Hill-type muscle model is used to describe muscle-tendon dynamics and the dependence of muscle force on muscle fiber length and velocity. Skeletal motion is modeled with Newtonian rigid body dynamics and smooth approximations of compliant Hunt-Crossley foot-ground contacts.

$$
\begin{aligned}
J = {} & w_1 \int_{t_0}^{t_f} \sum_{i=1}^{N_m} a_i^2(t)\, dt + w_2 \int_{t_0}^{t_f} \sum_{j=1}^{N_{\text{tm}}} e_{\text{tm},j}^2(t)\, dt \\
& + w_3 \int_{t_0}^{t_f} \|\boldsymbol{q}(t) - \boldsymbol{q}_{\text{ref}}(t)\|^2\, dt + w_4 \int_{t_0}^{t_f} \|\dot{\boldsymbol{q}}(t) - \dot{\boldsymbol{q}}_{\text{ref}}(t)\|^2\, dt \\
& + w_5 \int_{t_0}^{t_f} \|\ddot{\boldsymbol{q}}(t) - \ddot{\boldsymbol{q}}_{\text{ref}}(t)\|^2\, dt
\end{aligned} \tag{1}
$$

where t_0 and t_f are the initial and final times, w_i with $i = 1, \ldots, 5$ are weights, t is time, N_m is the number of muscles, and N_{tm} is the number of torque motors. Experimental joint positions, velocities, and accelerations are low-pass filtered using fourth-order, zero-lag Butterworth filters (default cutoff frequencies are 12 Hz for gait trials and 30 Hz for non-gait trials). Each cost term is scaled with empirically determined weights. To avoid singular arcs , a penalty function is appended to the cost function with the remaining control variables. Note that the optimal control problem formulation can be tailored to the activity of interest to incorporate activity-based knowledge by, for instance, adjusting the cost function, constraints, and filter settings. The optimal control problems are formulated in Python with CasADi (v3.5), using direct collocation and implicit formulations of the muscle and skeletal dynamics. Algorithmic differentiation is used to compute derivatives, and IPOPT is used to solve the resulting nonlinear programming problems with a convergence tolerance of 1×10^{-3} (all other settings are kept at default).

3. Surrogate model

To train the surrrogate model we use Uhlrich et al. (2022) to generate training data. We use $L1Smooth$ loss between the predicted and simulationed muscle activations

Metrics	R2	RMSE
Soleus (Left)	0.639	0.117
Soleus (Right)	0.614	0.119
Vastus Intermedius (Left)	0.665	0.068
Vastus Intermedius (Right)	0.676	0.063
Vastus Lateralis (Left)	0.873	0.112
Vastus Lateralis (Right)	0.799	0.134
Vastus Medialis (Left)	0.727	0.089
Vastus Medialis (Right)	0.679	0.092

Table 1: Metrics for Surrogate

4. Baselines

We utilize the joint representation Guo et al. (2022), used in other works recent human motion generation (Tevet et al. (2022), Zhang et al. (2023)), where the pose at each instant is represented as a 263-dimensional pose vector composed of various components: $\mathbf{p} = [\mathbf{r}_r, \mathbf{r}_l, \mathbf{r}_y, \mathbf{r}_i, \mathbf{r}_o, \mathbf{v}_l, \mathbf{f}_c]$, where $\mathbf{r}_r \in \mathbb{R}^{B\times N\times 1}$ represents the root rotation velocity, $\mathbf{r}_l \in \mathbb{R}^{B\times N\times 2}$ is the root linear velocity, $\mathbf{r}_y \in \mathbb{R}^{B\times N\times 1}$ is the y-coordinate of the root, $\mathbf{r}_i \in \mathbb{R}^{B\times N\times (J-1)\times 3}$ is the rotational information for the remaining $J-1$ joints (excluding the root joint), $\mathbf{r}_o \in \mathbb{R}^{B\times N\times (J-1)\times 6}$ is the rotation data for the remaining $J-1$ joints, $\mathbf{v}_l \in \mathbb{R}^{B\times N\times J\times 3}$ is the local velocity for all J joints, and $\mathbf{f}_c \in \mathbb{R}^{B\times N\times 4}$ is the foot contact information.

5. Hyperparmeters

References

Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3d human motions from text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5152–5161, 2022.

Argument	Default Value	Description
topk	10	Choose the best topk results from the total number of samples
loss-temporal	10	Hyperparameter for temporal regularizer
loss-proximity	0.1	Hyperparameter for proximity regularizer
loss-foot	1.0	Hyperparameter for foot regularizer
feet-threshold	0.01	Hyperparameter in meters to calculate whether joint is in contact with the ground

Table 2: Hyperparameters for surrogate guidance

Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H Bermano. Human motion diffusion model. *arXiv preprint arXiv:2209.14916*, 2022.

Scott Uhlrich, Antoine Falisse, Łukasz Kidziński, Julie Muccini, Michael Ko, Akshay Chaudhari, Jennifer Hicks, and Scott Delp. Opencap: 3d human movement dynamics from smartphone videos. 07 2022. doi: 10.1101/2022.07.07.499061.

Jianrong Zhang, Yangsong Zhang, Xiaodong Cun, Shaoli Huang, Yong Zhang, Hongwei Zhao, Hongtao Lu, and Xi Shen. T2m-gpt: Generating human motion from textual descriptions with discrete representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.