Thesis Report 2 : 23 February - 02 March

Goals

- Download and prepare Emotional Mocap Data ✅
- Convert Mocap Data BVH to Deepmimic Friendly format ✅
- Define which LMA features to extract ✅
- Figure out and implement way to extract data from running physics model (policy controlled) ✅
- Figure out and implement way to extract data from running kinematic model (mocap data) ✅
- Convert extracted data into LMA features ✅
- Have data files ready to be used to train LMA-Emotion mapping NN ✅

Last Week Leftovers:

- Install stuff on remote server ✅
- Perform Spacetime Bounds training on remote server ✅

Done

- Sent email to SLICE server admins in order to install Spacetime Bounds dependencies on remote server
- Properly installed and prepared Spacetime Bounds to be ran on remote server (i.e ran Make command and installed remaining Python dependencies)
- Ran Spacetime Bounds training on remote server
- **Figured out how to extract joint and link data from running simulation model on PyBullet**
- **Figured out how to extract joint and link data from running kinematic model on PyBullet**
- **Created a Class - LMAExtractor**
    - This class is capable of extracting the joints and link data from the running model (be it kinematic or physics-based).
    - It extracts this data on each frame (we must call function *record_frame* on the While cycles responsible for leaving the simulation running)
    - Internally stores link and joint data in a dictionary with the format:

```
{
frame: [frame_data in deepmimic format],
link_name: [link_position (3D), link_rotation (4D), joint_velocity (1D or 3D)]
}
```

- ◦ Data of the following links is stored:

```
root: [link_position (3D), link_rotation (4D), [linear_velocity (3D), ang_velocity (
chest: [link_position (3D), link_rotation (4D), linear_velocity (3D)]
neck: [link_position (3D), link_rotation (4D), linear_velocity (3D)]
right_hip: [link_position (3D), link_rotation (4D), linear_velocity (3D)]
right_knee: [link_position (3D), link_rotation (4D), linear_velocity (1D)]
right_ankle: [link_position (3D), link_rotation (4D), linear_velocity (3D)]
right_shoulder: [link_position (3D), link_rotation (4D), linear_velocity (3D)]
right_elbow: [link_position (3D), link_rotation (4D), linear_velocity (1D)]
right_wrist: [link_position (3D), link_rotation (4D), linear_velocity (0D)]
left_hip: [link_position (3D), link_rotation (4D), linear_velocity (3D)]
left_knee: [link_position (3D), link_rotation (4D), linear_velocity (1D)]
left_ankle: [link_position (3D), link_rotation (4D), linear_velocity (3D)]
left_shoulder: [link_position (3D), link_rotation (4D), linear_velocity (3D)]
left_elbow: [link_position (3D), link_rotation (4D), linear_velocity (1D)]
left_wrist: [link_position (3D), link_rotation (4D), linear_velocity (0D)]
```

- ◦ Data is then converted into LMA features every X frame (where X can be specified but, by default, is every 30 frames)
- ◦ LMA data is computed in the following format and either returned or written to a file directly:

```
{
    frame_counter: frame at which LMA features were computed,

    label: PAD Emotional Coordinates (3D)

    lma_features: [
        average hand_distance (1D),
        average l_hand_hip_distance (1D),
        average r_hand_hip_distance (1D),
        average feet_distance (1D),
        average l_hand_chest_distance (1D),
        average r_hand_chest_distance (1D),
        average l_elbow_hip_distance (1D),
        average r_elbow_hip_distance (1D),
        average chest_pelvis_distance (1D),
        average neck_chest_distance (1D),

        average neck_rotation (3D)
        average pelvis_rotation (3D)
```

```
        std l_hand_positions (1D)
        std r_hand_positions (1D)

        average l_forearm_velocity (1D)
        average r_forearm_velocity (1D)
        average pelvis_velocity (3D)
        average l_foot_velocity (3D) # May end up not being used
        average r_foot_velocity (3D) # May end up not being used

        average upper_body_volume (1D)
        average total_body_volume (1D)
        average distance_traveled (1D) # May end up not being used
    ]
}
```

- Downloaded several mocap data BVH files from http://dancedb.cs.ucy.ac.cy/main/performances
- Downloaded several mocap data BVH files from https://motion.hacettepe.edu.tr/?c=locomotion#
- Downloaded several mocap data BVH files from https://physionet.org/content/kinematic-actors-emotions/1.0.0/
- Created a script that mass converts all BVH files in given folders into Deepmimic mocap friendly files
- **Manually tweaked BVH files in order to allow them to be gracefully converted into Deepmimic Mocap format (tweaked skeletons, scales and rotations and messed with the BVHConversion library source code)**
    - **Note**: Even having done this, there are conversion artifacts. See Problems section
- Created a script that takes all Deepmimic Mocap data in a given folder and extracts all LMA features into files (to be used as data for training)
- **Got the training data to start creating our Neural Networks!**

Left Undone

Nothing

Problems

- **[PARTLY SOLVED]** Mocap of locomotion with emotions has a weird artifact which makes the entire skeleton tilted. I've tried multiple ways of solving this, either by messing with the core skeleton's rotations and offsets in the BVH file or even altering the code of the library that converts BVH to a Deepmimic friendly format, but to no avail. https://imgur.com/a/mVa15Ci

    - **[PARTIAL SOLUTION]** This data will only be used to extract LMA features in order to train our LMA-Emotion mapping. Since LMA features are correlations between the body and itself, this may not be an issue

- **[PARTLY SOLVED]** Mocap data from https://physionet.org/content/kinematic-actors-emotions/1.0.0/ is unusable (at the moment) due to the skeleton they used being much more complex than what Deepmimic is ready for.

    - **[PARTIAL SOLUTION]** The actual problem was due to their skeleton having one less child joint on the feet (rather than what I previously thought was the problem - too complex of a skeleton). As such, I simply altered the source code of the BVHToDeepmimic library in order to account for this. The feet are kind of incorrect due to this, but at least the overall animation seems to be working. - https://imgur.com/a/0YS9480

- **[NOT SOLVED]** I fear that the data we have won't be enough to train our neural networks to go from the LMA features to the PAD model. This concern stems from the fact that the PAD model uses a 3D axis system, discerning emotions by their P, A and D coordinates. As such, we would need emotional representatives of **AT LEAST** all extreme emotions (i.e emotions with the coordinates (0,0,0), (0,0,1), (0,1,0), (0,1,1), (1,0,0), (1,0,1), (1,1,0) and (1,1,1)) which we do not currently have.

    - **[POSSIBLE SOLUTION]** A possible solution would be to instead use a discrete set of emotions (Neutral, Sad, Happy, Angry, Afraid, Excited, ...) and allow the user to swap between those, rather than allowing them to specify PAD coordinates. Obviously this is not ideal, but due to lack of data, this might be the only solution. Nevertheless I will begin by training the networks to map to PAD coordinates and will check their effectiveness

Notes

Thoughts

The week went more or less smoothly. Although I didn't work as much as I wanted (as I was feeling ill during the weekend/monday), I still managed to do all I set out to do and even account for all the issues the mocap datasets are causing me. Gained more insight as to how Spacetime bounds is integrated with PyBullet, learned how to actually use PyBullet (https://raw.githubusercontent.com/bulletphysics/bullet3/master/docs/pybullet_quickstartguide.pdf) and started to fully comprehend how BVH files are structured and how the BVHToDeepmimic library works (having been able to alter it and work with even the most unfriendly of data).

The current number one concern is, nevertheless, the lack of data. I'm thinking of alternatives, should I not be able to actually perform the mappings to the PAD model. The two current ideas would be to downgrade to RCM (since its easier to find all representative emotions of that system),

or use a discrete set of emotions instead (which may prove to yield better overall results, albeit making the emotional tweaking system more limited)

Work Hours

This week I worked everyday from about 1pm-6pm Didn't work Sunday because I was feeling ill (possible food intoxication)