Myo the Force Be With You

Karen Yang *, David Pan †
Stanford EE 267, Virtual Reality, Course Report, Instructors: Gordon Wetzstein and Robert Konrad

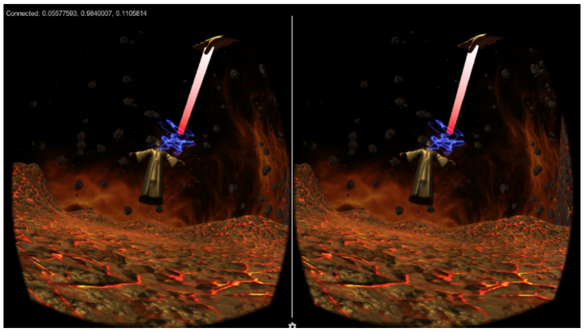

Figure 1: *Screenshot of the game.*

Abstract

Inspired by the Force from Star Wars, we developed a Telekinesis game that allows the player to equip the force to pick up and control remote objects. We used Myo Armband to provide 9-axis IMU spatial data and EMG gestural data. In Unity, we designed and implemented an arm model, physics model, and push-pull gestures recognition to bestow the player a realistic and natural experience of applying the force within the virtual world. Our demo presents a potential natural interaction paradigm with Kinematics and EMG data sensor fusion for future VR and AR applications.

1 Introduction

EMG data is already widely applied in assistive or medical wearable devices for the use of limb functions, gait controls, breathing patterns and in controlling a rehabilitation robotic arm, etc. Several advantages of surface EMG and IMU data fusions include the use of natural biosignals that make the user feel more natural compared to handlers, and its minimal requirement for moving range and lighting avoids the space limitation and occlusion problems. It provides the potential of adding on multiple degrees of freedom by adding multiple sensors, or estimating of the force being exerted during grasping from the electrical motors noise, allowing for many possibilities of movement detection of a single hand or the full body. Current literatures shows that there is not yet a full understanding of EMG pattern. EMG can also vary among different people with different age, body fat and skin condition.

In our project, we experimented with the fusion of EMG data combined with IMU data, enabled by Myo armband, to implement a game control. The data is sent to the Unity program via bluetooth, allowing a wireless control, and allows for the rotation angle calculation and simple gesture classification.

2 Related Work

Haque et al. developed a point-and-click system for 2D displays using the Myo armband. The orientation of Myo determines the location of the mouse pointer, and Myo hand gestures are used to click. In our project, instead of mapping the orientation of the Myo onto a 2D plane, we map it onto a sphere.

Nymoen et al. developed a musical instrument using the Myo. To evaluate the quality of the acceleration data from the Myo, they added reflective markers on the Myo and used an optical motion tracking system to obtain more accurate acceleration data. They found that while some areas had some small differences, the peaks in the acceleration data were properly aligned. This helps justify our peak-detection algorithm for push-pull gestures.

3 Approach

3.1 Controls

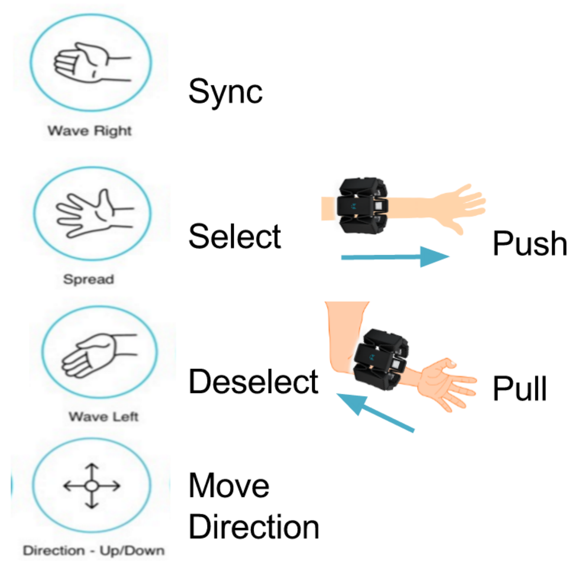

Figure 2: *Controls for the game.*

- Synchronize - calibrate the arms position using the "wave out" gesture
- Select - select an object using the "spread fingers" gesture
- Deselect - deselect an object using the "wave in" gesture
- Move - move the object along a sphere by moving the arm
- Push or pull - push or pull the object by making push or pull movements

3.2 Arm Model

Virtual Arm is the key element in our game design. We computed the shoulder's position, as well as orientation relative to the eye (Main Camera). By defining a syncing gesture (wave out), we enable players to update the arm position relative to the HMD at the beginning of the game, so that the position of the virtual arm will appear in the world view as if it is in the real arm position.

*e-mail:kaiyuany@stanford.edu
†e-mail:napdivad@stanford.edu

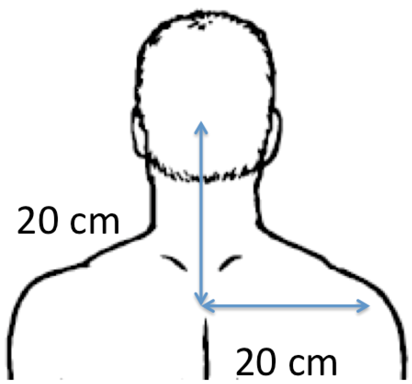

Figure 3: *Dimensions used to compute the shoulder position.*

Since the Myo only provides the orientation of the forearm but not the position, we modeled the arm as a stick that rotates about the shoulder position. We used the typical arm length of 64 cm as the stick length. To compute the shoulder position relative to the eye position, which is essentially the camera position, we used the typical eye-to-shoulder vertical distance of 20 cm and the typical shoulder-to-shoulder distance of 40 cm. The vector formula is the following:

$$\mathbf{shoulderPosition} = \mathbf{eyePosition} - (0, eyeToShoulderDistance, 0) + Rotate(\mathbf{eyeDirection}, 90) * 0.5 * shoulderToShoulderDistance$$

3.3 Physics Model

We define the following variables:

- $\mathbf{z}$ is the position of the arm.
- $\hat{\mathbf{z}}_\theta$ is the unit direction vector of the arm.
- $\mathbf{x}$ is the position of the object that the user is controlling.
- $\dot{\mathbf{x}}$ is the velocity of the object that the user is controlling.
- $\mathbf{y}$ is the desired position of the object (i.e. where the user wants the object to be).
- $\mathbf{F_r}$ is a force that pushes or pulls the object away from or towards the user. It is activated by the push and pull gestures.
- $\mathbf{F}_\theta$ is a force from $\mathbf{x}$ to $\mathbf{y}$. It pushes the object towards the desired position.
- $\mathbf{F_d}$ is a damping force that depends on the velocity of the object.
- α is a constant scaling factor for $\mathbf{F_r}$
- β is a constant scaling factor for $\mathbf{F}_\theta$
- γ is a constant scaling factor for $\mathbf{F_d}$
- $\mathbf{F}$ is the net force applied to the object. It is the sum of $\mathbf{F_r}$, $\mathbf{F}_\theta$ and $\mathbf{F_d}$.

We made a physics model where the selected object is guided by a force to where the user wants it to be. The users desired position for the object is assumed to be along the surface of a sphere with its center being the users position and its radius being the distance between the user and the object. The orientation of the users arm determines the unique desired position on the surface of this sphere.

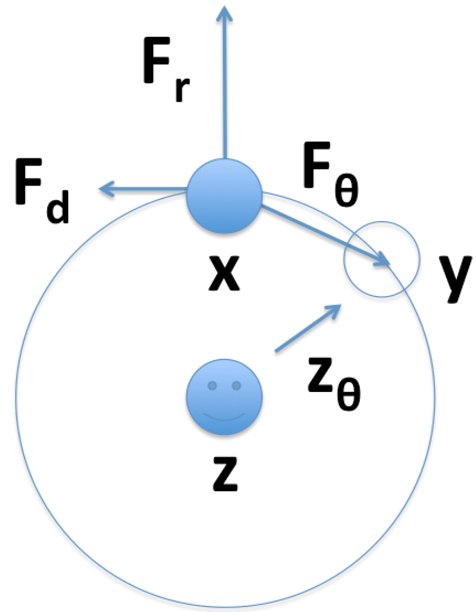

Figure 4: *Force diagram for an object at position* $\mathbf{x}$.

An angular force pointing from the current object position to the desired object position is applied to the object. In addition, if the user makes a push or pull gesture, a corresponding radial force is applied to the object.

Using a mass-spring-damper model [Becedas and Sira-Ramirez 2007] as inspiration, the magnitudes of the angular and radial forces are directly proportional to displacement. The angular force is scaled by the distance between the current object position and the desired object position, and the radial force is scaled by the distance between the user and the object. To prevent the object from oscillating too much, a damping force is applied to the object, where the damping force is a scaled version of the object velocity pointing in the opposite direction.

The following are the equations for the physics model:

$$\begin{aligned}
\mathbf{F} &= \mathbf{F_r} + \mathbf{F}_\theta + \mathbf{F_d} \\
\mathbf{y} &= \|\mathbf{x} - \mathbf{z}\| \, \hat{\mathbf{z}}_\theta + \mathbf{z} \\
\mathbf{F_r} &= \pm\alpha \frac{\mathbf{x} - \mathbf{z}}{\|\mathbf{x} - \mathbf{z}\|} \\
\mathbf{F}_\theta &= \beta \frac{\mathbf{y} - \mathbf{x}}{\|\mathbf{y} - \mathbf{x}\|} \\
\mathbf{F_d} &= -\gamma \dot{\mathbf{x}}
\end{aligned}$$

3.4 Push and Pull Gestures

We implemented push and pull gestures using the Myo's acceleration data and orientation quaternion. A push is when the Myo is thrust forward, and a pull is when the Myo is thrust backward. We computed the component of linear acceleration along the Myos forward direction, and we used a peak-detection algorithm to determine whether the gesture was a push or a pull.

To compute the component of linear acceleration, the Myo orientation quaternion is applied to the acceleration in the sensor frame to obtain the acceleration in the world frame. The gravity vector (0, -1, 0) is subtracted from the result to obtain the linear acceleration in the world frame. Then the dot product of this result and the unit direction vector of the Myo yields the component of linear acceleration along the Myos forward direction. The formula is

$$a = (\mathbf{Q} * \mathbf{a_s} - \mathbf{g}) \cdot \hat{\mathbf{z}}_\theta$$

where $\mathbf{Q}$ is the Myo's orientation quaternion, $\mathbf{a_s}$ the acceleration in the sensor frame, $\mathbf{g}$ the gravity vector, and $\hat{\mathbf{z}}_\theta$ the Myo's unit direction vector.

The peak-detection algorithm works as follows: if there is a negative peak followed by a positive peak, then there is pull; if there is a positive peak followed by a negative peak, then there is push. Figure 5 shows the acceleration signal for a pull and a push.

Figure 5: *Component of linear acceleration along the Myo's forward direction. The first pair of peaks corresponds to a pull, and the second pair corresponds to a push.*

4 Evaluation

The arm model works well and the physics model of the Force feels natural with the damping effect. The physics model makes heavy objects such as the spaceship actually "feel" heavy because they are harder to move.

Push and pull gestures were not as stable as expected. The acceleration data appeared to be noisy, and we will need to further improve robustness in our peak-detection algorithm.

Some Myo gestures were not recognized consistently. This appears to be a problem with the Myo; this issue was brought up in [Nymoen and Jensenius 2015].

5 Discussion

Myo provides a nice package of a 6-DOF IMU data and EMG data all packaged with bluetooth interface. However, the precision for its EMG data is very limited, and the position tracking is not possible. Thus our arm model, physics model, and gestures were constrained by the Myo. We wish to further integrate the EMG sensors with more accurate gesture classification using Machine Learning, and potentially add a depth camera to aid the position tracking and the gesture recognition.

We have a functioning Extended Kalman Filter implemented in Matlab and in FPGA (for another class project). The covariance matrix of this model has been simplified and optimized for the Myos application. We did not have time to implement the streaming of the Myos raw data into the Kalman Filter and we realize that this might slow down the demo, but the integration of Kalman Filter calibration will be of interest in the future to synchronize arm position a lot because of gyro drift.

Acknowledgements

Special thanks to Professor Gordon Wetzstein and Robert Konrad, other staff in the Stanford EE Department, and all parties that supported EE 267.

References

BECEDAS, J., M. G. F.-B. V., AND SIRA-RAMIREZ, H., 2007. Algebraic identification method for mass-spring-damper system.

GEORGI, M., C. A., AND SCHULTZ, T. 2015. Recognizing hand and finger gestures with imu based motion and emg based muscle activity sensing. In *Proceedings of the International Conference on Bio-inspired Systems and Signal Processing*.

HAQUE, F., N. M., AND VOGEL, D., 2015. Myopoint: Pointing and clicking using forearm mounted electromyography and inertial motion sensors.

NYMOEN, K., H. M., AND JENSENIUS, A., 2015. Mumyo evaluating and exploring the myo armband for musical interaction.