Designing and Fitting FES and Prosthetic Systems in a Virtual Reality Environment

Markus Hauschild, Rahman Davoodi and Gerald E. Loeb
Department of Biomedical Engineering, University of Southern California
markus.hauschild@usc.edu

Abstract

Building and testing novel prosthetic limbs and FES control algorithms is expensive and risky. Here we describe a Virtual Reality Environment (VRE) to facilitate and accelerate the development of novel systems. In the VRE subjects/patients can operate a simulated limb to interact with virtual objects. Realistic models of all relevant musculoskeletal and mechatronic components allow the development of entire prosthetic systems in VR before introducing them to the patient. The system is used both by engineers as a development tool, and by clinicians to fit prosthetic devices to patients.

I. INTRODUCTION

Spinal cord injury (SCI) patients frequently are left with a partially paralyzed arm, whereas amputees have lost the limb itself. In both cases patients are severely handicapped, not being able to perform many of the simple activities of daily life. *Functional Electrical Stimulation* (FES) is a promising approach to reanimate the paralyzed muscles of SCI patients [1], whereas amputees can be helped with powered arm-hand prostheses. Both technologies are still active research topics, however, and no widely accepted solutions are on the market. No reliable algorithms to control arm-hand coordination exist, and any such algorithms are likely to require extensive customization and fitting for each individual patient [2]. Being able to design truly functional systems for both groups of patients would be a major breakthrough, but building and testing such devices is expensive, risky and time-consuming.

In order to facilitate and accelerate the development, we designed a *virtual reality environment* (VRE) in which subjects can operate a simulated arm system to interact with virtual objects.

This paper describes its architecture and how we use it throughout the design process for FES and prosthetic devices, starting with prototyping of software and hardware components, all the way to fitting the device to the patient and patient training. Currently the VRE is actively used by a number of research groups, and the goal is to introduce a user-friendly version to clinicians in the field of musculo-skeletal rehabilitation, post-stroke motor rehabilitation, prosthetology, etc.

II. THE VR ENVIRONMENT - OVERVIEW

Fig. 1 depicts a typical configuration for a reaching experiment. The subject performs a reaching movement to a virtual target, and a motion capture system tracks the arm movement. Real time algorithms determine the resulting virtual arm trajectory. A 3D head mounted display (HMD) provides visual feedback of the animated arm from the subject's perspective.

Fig. 2 shows the interaction and signal flow between subject/patient, control code, virtual arm, and feedback hardware. Typically the subject wears multiple sensors (e.g. motion tracking, EMG, head tracking, ...) which provide the input to the real-time control code. The output of that control code drives the virtual limb actuators, which interact with the dynamic model of the limb to cause movement, which is displayed as a 3D animation. Haptic feedback hardware (robots, tactile arrays, ...) may be used to provide additional feedback.

During the early design phase of an advanced prosthetic device it will be unclear what features will be incorporated in the final design. It is efficient, therefore, to design and test components in VR, where design iterations can be performed easily without discarding previous work. Such a configuration is also

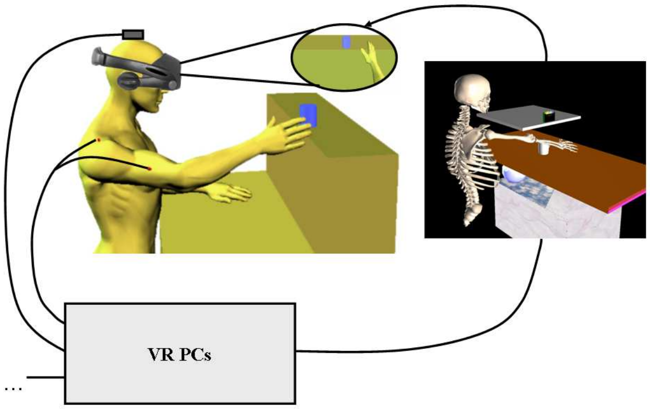

Fig. 1. In the VR environment subjects can operate a simulated prosthetic arm to interact with virtual objects. Multiple input modalities such as motion tracking systems, EMG/EEG electrodes, etc. provide maximum flexibility when evaluating different control approaches.

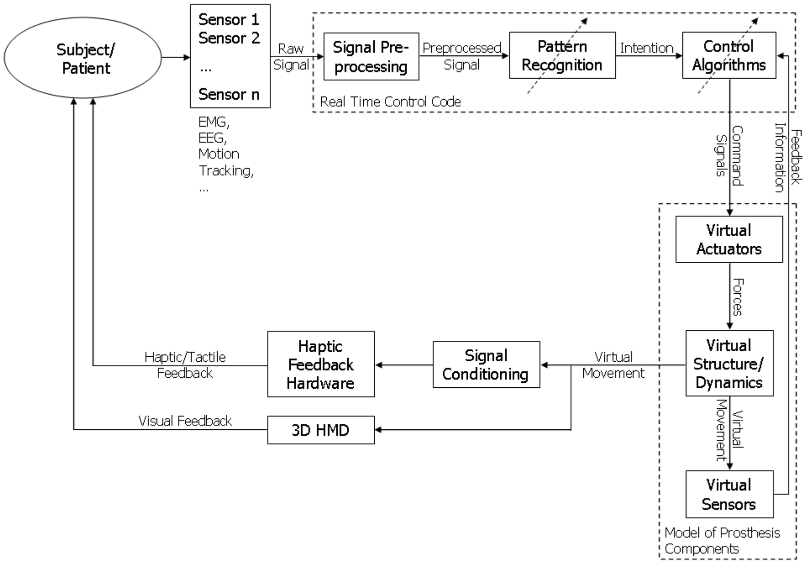

Fig. 2. High level diagram of the VR environment, used for prosthesis design and early patient training.

helpful for the early patient training phase. The level of complexity can be increased gradually, and the parameters can be adjusted to correct for errors and limitations. Finally, once the patient is confident with the prosthesis' use, the control algorithms can be implemented on an embedded CPU that is part of the prosthetic device.

A. *Hardware Configuration*

The VRE as shown in Fig. 3 consists of 3 PCs and multiple input/output hardware components. We use a magnetic motion tracking system (Flock of Birds, Ascension Technology Corporation), to capture arm movements, but the VRE system can accept properly formatted data from any motion capture technology. A gyro-based 3-axis sensor (3DM-GX1, Microstrain) captures head movements. EMG electrodes and additional inputs can be connected to a general purpose data acquisition board (PCI-6040E, National Instruments). The real-time xPC samples all these inputs, executes the prosthetic arm control code, and runs real-time dynamics of musculoskeletal and mechatronic arm components.

The resulting output is sent to visualization PCs for animation of the model limb. Multiple PCs can be used for visualization. Our configuration employs one PC for subject visualization via 3D head mounted display (HMD) (NVISOR SX, NVIS), and a second PC for operator visualization. The subject PC provides a stereoscopic view of the ongoing experiment to the subject, whereas the operator PC only displays a 2D view for experiment supervision purposes. Simultaneously the operator PC provides a user interface for experiment control, online parameter tuning, etc. The operator PC is also used for code development, and to up- and download files to and from the real-time PC.

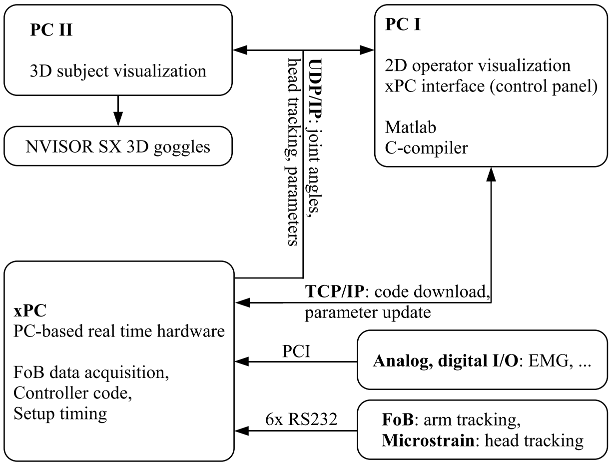

Fig. 3. VR configuration - block diagram. The system consists of subject PC, operator PC, the real time PC (xPC) and a motion tracking system for head and arm movements.

B. *Software Configuration*

All control and dynamics related algorithms were developed in SIMULINK. Many powerful tools for mechanical dynamics (e.g. SimMechanics), visualization and control are already available on this platform. SIMULINK itself is not real-time capable, but the xPC-Target toolbox allows execution of SIMULINK code on a separate real time PC (the xPC) with minor modifications only. MSMS (Musculoskeletal Modeling Software), a Java-based program,developed in our lab [3], [4], is used to visualize the animated arm in a virtual environment for both subject and operator.

III. REAL TIME IMPLEMENTATION

Fig. 4 shows the top level of the real-time implementation in SIMULINK. The hardware input block on the left side contains code related to data acquisition. Currently it drives a 5-sensor 6 DoF FoB system for arm tracking, a 3 rotational DoF MicroStrain sensor for head tracking, 16 analog inputs for EMG acquisition, and 8 digital inputs (e.g. for external triggers, control switches, etc).

The hardware output block on the right side of Fig. 4 contains code to transmit data from the internal real-time processes to the outside world. This includes a network interface to broadcast joint and head-tracking angles to external visualization PCs, 2 analog outputs, and multiple digital outputs. The hardware output block also contains code for data logging. Entire experiments can be recorded to hard drive for later analysis and playback.

Most real-time code is located between hardware input and output layer. In Fig. 4 only arm angle extraction and head tracking blocks are shown. These blocks convert the motion tracking data from a sensor coordinate system based format to a visualization software (MSMS) compatible format. The configuration shown in Fig. 4 simply tracks the subject's arm motions, and drives the VR animation in MSMS, i.e. the VR animation reproduces the movements the subject actually performs. To simulate control of a prosthetic or FES limb, one can add dynamics and control blocks to the signal flow instead of sending the transformed motion-capture data to the output block directly.

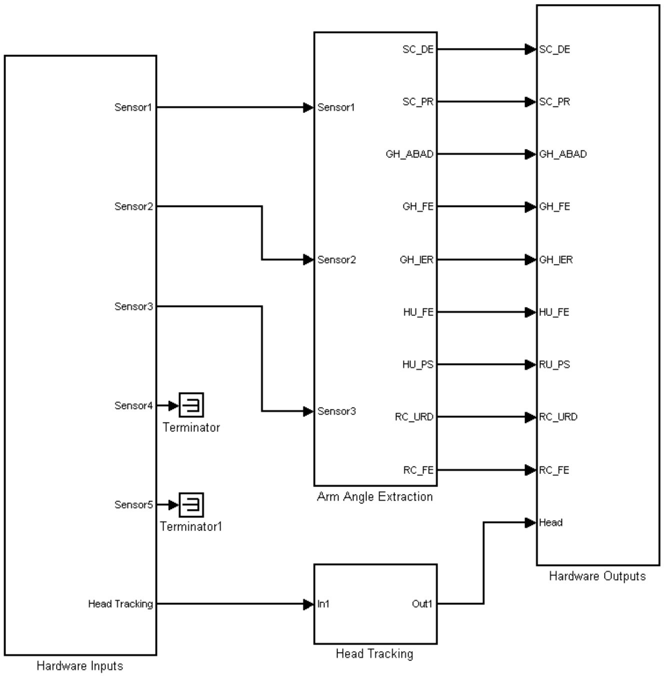

Fig. 4. The main screen of the real time simulation environment in SIMULINK showing hardware input layer, signal processing and control layer (here: angle extraction for arm and head-tracking), and hardware output layer.

IV. VISUALIZATION

As shown in the VR setup diagram (Fig. 3), the output of the real time PC consists of angles describing arm and head movements, updated in real time. These data are transmitted via UDP/IP network connection to the multiple visualization PCs, allowing the subject/patient to obtain 3D visual feedback, and the operator or physician to monitor the experiment.

We are using Musculoskeletal Modeling Software (MSMS) to animate a human skeleton based on the transmitted angles within a virtual experimental environment. This software was developed primarily to provide clinicians with a tool to fit BION-based FES solutions to individual patients with a wide range of sensorimotor disabilities. It consists of a basic graphic user interface to load, manipulate, and save musculoskeletal models that can then be created in MSMS or imported directly from SIMM [5], an older model creation system. The MSMS models can be animated by motion data from a saved file, by motion data streamed in real-time from a motion capture system, or by motion data computed by a dynamics engine.

V. MOTION TRACKING

The need for and selection of motion tracking technology depends on the application and is independent of the VRE. The limb to be controlled will generally be equipped with position sensors whose information can be sent to the VRE once it is transformed into the appropriate coordinates as described below.

We use the Flock of Birds (FoB) (Ascension Technology Corporation) magnetic motion tracking system to track shoulder, elbow and wrist in real time. The FoB system uses sensors, each housing three orthogonally located coils, wound on ferrite material, and moving within a low frequency electromagnetic field (thus avoiding eddy currents) generated by a second three-coil source (transmitter) [6]. The sensors are attached to the limbs to be tracked, whereas the transmitter remains stationary.

We are relying on a magnetic motion tracking system for the following reasons [7]:

- magnetic motion tracking provides position and orientation information
- a minimum effort for calibration is necessary (only source position and orientation need to be known)
- the technology is mature and robust
- it is real time capable
- magnetic motion tracking is less expensive than optical tracking.

However, magnetic motion tracking suffers from the following limitations:

- sensitivity to metal (it has been shown [8] that mild steel produces significant interference, other metals don't distort the signal)
- limited range
- significant sensor noise.

We can overcome most of these limitations by optimizing certain aspects of the hardware setup: installing the transmitter near the intended workspace reduces sensor noise, avoiding mild steel in the environment, and digital filtering eliminates most interference.

A. *Arm Tracking*

We use three FoB sensors attached over the clavicle, humerus and radius for motion tracking (Fig. 5). Our model of the human arm currently consists of three segments with 7 degrees of freedom (DoF) (Fig. 6). Two translational DoF's at the shoulder were modeled as a rotary joint with two rotational DoFs (sternoclavicular joint). Three rotational DoFs at the shoulder were modeled as a 3 DoF rotary joint between the clavicle and humerus (GH joint). The ulna and radius were treated as one segment with two rotational DoFs at the elbow (HU joint) to account for elbow flexion/extension and wrist pronation/supination.

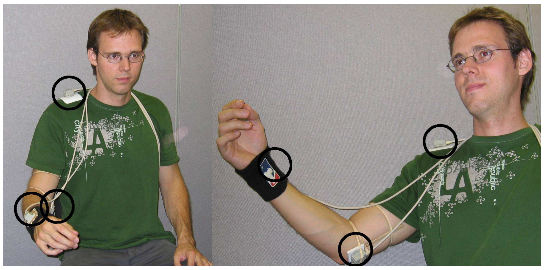

Fig. 5. Three sensors track the subject's right arm.

B. *Flock of Birds - Data Acquisition*

The arm tracking sensors are sampled at $100Hz$. To drive our VR animation we are currently using the sensors' orientation matrix output only, not their position output.

Communication between FoB and the xPC is handled entirely by code in the "Hardware Inputs" block (Fig. 4). It handles FoB initialization, collects the stream of raw sensor data, and restores the orientation matrices from the raw bitstream. These outputs are transmitted from the "Hardware Inputs" block to the "Arm Angle Extraction" block for further processing.

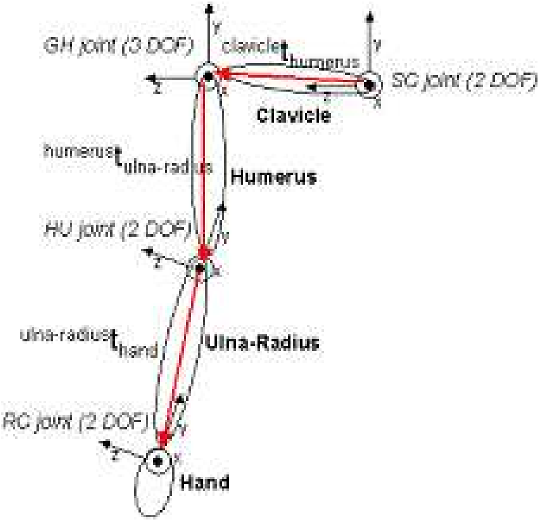

Fig. 6. Kinematic model of the human arm. The model has four segments (Clavicle, Humerus, Ulna-Radius and Hand) and four joints between them.

joint name	proximal segment	distal segment	rotational DoFs (in the order of Euler angle interpretation)
SC joint	Sternum	Clavicle	X: Depression/Elevation Y: Protraction/Retraction
GH joint	Clavicle	Humerus	X: Abduction/Adduction Z: Flexion/Extension Y: Iternal/External Rotation
HU joint	Humerus	Ulna-Radius	Z: Flexion/Extension Y: Pronation/Supination

TABLE I

JOINTS AND ROTATIONAL DOFS.

C. Arm Angle Extraction

The rotation matrix output from the “Hardware Inputs” block describes the sensors’ orientation with respect to the transmitter. These angles cannot be used to animate the VR skeletal model because the expected inputs to animate the arm are Euler angles, describing the angle of a distal bone segment with respect to a proximal bone segment.

The required conversion consists of correcting the sensor rotation matrices for axis misalignment errors and initial angle offset, and then extraction of the required angles from the corrected rotation matrix.

SC, GH, and HU joint angles were extracted from the sensors’ rotation matrices based on the following simplifying assumptions [9]:

- SC joint: the model assumes that the SC joint has only 2 DoFs, however there is another rotational DoF about the longitudinal axis (z-axis in the MSMS model). This longitudinal rotation is not an independent DoF, instead it is correlated with other shoulder DoFs [10]. This DoF was not included in the arm model because it would be hard to utilize such a dependent DoF as a command signal source for control. However this rotational component is captured by the sensor attached to the clavicle, and in order to keep the sensor readings consistent with the underlying model, it has to be eliminated. This was done by decomposing the matrix that describes current clavicle orientation wrt initial clavicle orientation into its Euler angles, and then by eliminating the unwanted DoF from the Clavicle matrix.
- HU joint: the rotation about the longitudinal axis of the Humerus segment is usually underestimated

by the sensor due to skin movement [11]. It is more correctly reflected by the measurement of the distal sensor, which we use to animate the Humerus segment.

- HU joint: elbow FE and forearm PS are assumed to be orthogonal, although the angle is known to be in the range of $84° \ldots 94°$ [12].

D. Head Tracking

The head tracking sensor allows the subject/patient to control the field of view for the 3D HMD. It increases the level of immersion and avoids motion sickness [13]. The 3DM-GX1 sensor, sampled at $100Hz$, transmits the orientation in Euler angle format that can be used directly to control the field of view in the visualization software. Initial calibration is performed in the "Head Tracking" block (Fig. 4).

VI. EMG PROCESSING

A wide range of EMG electrodes can be connected to the National Instruments A/D converter hardware inputs. We use bipolar electrodes with built in pre-amplification from B&L Engineering. Due to the pre-amplification by factor 330 the signal to noise ratio is very good, and the electrodes can be connected to the A/D converter without additional amplification. This type of electrode can be used in surface EMG configuration, but a fine wire adapter is also available for intramuscular EMG recording. The EMG electrodes are sampled at $1kHz$ and a $300Hz$ cutoff frequency first order low pass filter eliminates aliasing artifacts. An electrical isolation amplifier (ISO122, Burr-Brown) isolates the patient electrically without compromising the EMG signal.

On the software side our most basic EMG processing circuit includes raw signal rectification and envelope extraction (basically heavy low-pass filtering) to detect the level of EMG activity. This is typical for myoelectric control algorithms in which muscles still under voluntary control provide command signals to the prosthetic system.

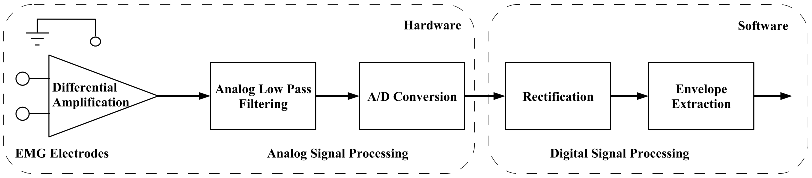

Fig. 7. EMG acquisition and signal conditioning.

VII. CONCLUSIONS AND FUTURE WORK

The VRE presented here can be used in different ways by researches and clinicians to design and fit novel prosthetic devices.

- Study of normal reaching will provide training sets and performance criteria for adaptive controllers.
- Feasibility analysis can be conducted entirely in VR to determine how a specific prosthesis will perform in a specific patient.
- Significant steps of control design can be performed in VR to identify performance and stability of candidate algorithms. This will be particularly important for adaptive control and neuronal networks that require many iterations.
- Patient training in VR will help the patient to master complexity in stages.
- Initial parameter tuning rules can be established in VR to allow autocalibration during active use.

The present system supports motion tracking and EMG inputs. Additional inputs can be incorporated easily. A stereoscopic view of arm and environment provides realistic visual feedback to the subject. Dynamic head tracking extends the usable extrapersonal workspace when using a head mounted display with limited field of view.

Currently the VRE is used for FES control design. FES control can be used to reactivate muscles that have been paralyzed due to spinal cord injury [1]. Synergies among various joints are known to exist for reaching movements [14], for example between shoulder and elbow. We intend to apply these synergies to reconstruct the elbow command signal from residual voluntary motion (in this case shoulder movements) [15].
We will use the VRE to

- record data and analyze synergies between joints while normal subjects perform reaching movements.
- train artificial neural networks to predict distal joint angles from proximal joint input.
- simulate FES arm movement, considering musculoskeletal dynamics [16], [17].

The VRE will also assist us in designing prosthetic arm-hand systems. Conventional control concepts such as digital switches, single channel, and dual channel EMG are implemented. Classifier based EMG control [18] will be supported in the near future.
We are aiming for a comprehensive approach, in which the VRE will serve as a platform to test a large number of hardware, software, and control concepts. Many of these may prove to be non-useful, others will improve function but perhaps only in particular combinations. Finding these through manual implementation of all possible combinations on real prosthetic hardware with real patients would be simply impossible.

REFERENCES

[1] G. E. Loeb and F. J. R. Richmond, *Neural Prosthesis for Restoration of Sensory and Motor Function*. CRC Press, 2000, ch. Implants for therapeutic and functional electical stimulation, pp. 75–99.

[2] D. J. Atkins, D. C. Y. Heard, and W. H. Donovan, “Epidemologic overview of individuals with upper-limb loss and their reported research priorities,” *JPO*, vol. 8, 1996, .

[3] R. Davoodi, C. Urata, and G. E. Loeb, “Muculoskeletal modeling software, MSMS,” Webpage: http://ami.usc.edu/msms/.

[4] R. Davoodi, C. Urata, E. Todorov, and G. E. Loeb, “Development of clinician-friendly software for musculoskeletal modeling and control,” in *Proc. of 26th Annual Intern. Conf. of the IEEE EMBS*, 2004.

[5] S. L. Delp and J. P. Loan, “A graphics-based software system to develop and analyze models of musculoskeletal structures,” *Computers in Biology and Medicine*, vol. 25, pp. 21–34, 1995.

[6] R. J. Stone, “Position and orientation sensing in virtual environments,” *Sensor Review*, vol. 16, pp. 40–46, 1996.

[7] S. Dyer, J. Martin, and J. Zulauf, “Motion capture white paper,” Tech. Rep., 1995.

[8] A. D. Milne, D. G. Chess, J. A. Johnson, and J. W. King, G, “Accuracy of an electromagnetic tracking device: a study of the optimal operating range and metal interference,” *J. Biomech.*, vol. 29, pp. 791–793, 1996.

[9] J. Lee, “Virtual reality environment for human-in-the-loop simulation of fes reaching controllers,” Master’s thesis, University of Southern California, 2005.

[10] F. C. T. van der Helm, “Analysis of the kinematic and dynamic behavior of the shoulder mechanism,” *J. Biomechanics*, vol. 27, pp. 527–550, 1994.

[11] A. G. Cutti, G. Paolini, M. Troncossi, C. A., and D. A., “Soft tissue artefact assessment in humeral axis rotation,” *Gait & Posture*, vol. 21, pp. 341–349, 2005.

[12] H. E. J. Veeger, B. Yu, K. N. An, and R. H. Rozendal, “Parameters for modeling the upper extremity,” *J. Biomechanics*, vol. 30, pp. 647–652, 1997.

[13] G. C. Burdea and P. Coiffet, *Virtual Reality Technology*. John Wiley & Sons, Inc., 2003.

[14] S. D. Iftime, L. L. Egsgaard, and M. B. Popovic, “Automatic determination of synergies by radial basis function artificial neural networks for the control of a neural prosthesis,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 13, pp. 482–489, 2005.

[15] R. R. Kaliki, R. Davoodi, and G. E. Loeb, “Synergistic prediction of elbow flexion and extension using a multi-layer perceptron neural network,” in *28th IEEE-EMBS*, 2006.

[16] G. E. Loeb and C. Ghez, *Principles of Neural Science*. McGraw-Hill, 2000, ch. The Motor Unit and Muscle Activation, pp. 674–694.

[17] E. J. Cheng, I. E. Brown, and G. E. Loeb, “Virtual muscle: A computational approach to understanding the effects of muscle properties on motor control,” *J. of Neuroscience Methods*, vol. 101, pp. 117–130, 2000.

[18] K. Englehart, B. Hudgins, P. A. Parker, and M. Stevenson, “Classification of myoelectric signals using time-frequency based representations,” *Medical Eng. & Physics*, vol. 21, pp. 431–438, 1999.