A Real-Time Virtual Integration Environment for Neuroprosthetics and Rehabilitation

Robert S. Armiger, Francesco V. Tenore, William E. Bishop, James D. Beaty, Michael M. Bridges, James M. Burck, R. Jacob Vogelstein, and Stuart D. Harshbarger

We present a Virtual Integration Environment (VIE) for neuroprosthetic devices. The VIE is an integrated software framework for coordinating distributed engineering design efforts. It provides researchers with a means to implement and test all major components involved in developing and using a neuroprosthetic device, using both virtual and hardware-in-the-loop simulations. Data provided from neural signals in the motor cortex, nerve fibers of the peripheral nervous system, and surface or intramuscular myoelectric signals can be acquired and processed to provide real-time control of real or virtual prosthetic upper limbs. The VIE can therefore be used to visualize and monitor performance of various design approaches, evaluate neural signal analysis algorithms, simulate emerging mechatronic elements, train end users to control real or virtual neuroprosthetic devices, and configure and customize take-home devices. Here we provide a comprehensive description of the system, as well as a summary of its applications for myoelectric control and neural research at multiple academic institutions.

INTRODUCTION

Within the last decade, neural prosthetics have seen tremendous progress based on advances in neuroscience, robotics, controls, electrical engineering, and applied mathematics. Virtual and real devices can now be controlled with signals recorded from a variety of locations along the neural efferent pathway from the brain to the hand. These signals originate within and on the surface of the motor-controlling cortex of the brain, along peripheral nerves, within individual muscles, and superficially on the skin surface.[1–3] This diversity of biological signals requires a variety of specialized processing algorithms to appropriately decode each signal set. Furthermore, it is expected that signals from multiple modalities will soon be used simultaneously to provide accurate and robust decoding. Because of the growing complexity of neuroprostheses, a common framework is needed to support development and comparison of neuroprosthetic systems.

We have developed a Virtual Integration Environment (VIE) as part of the Defense Advanced Research Projects Agency (DARPA) Revolutionizing Prosthetics 2009 (RP2009) program to provide a common platform for neuroprosthetic research and development. The aim of RP2009 is to develop an advanced prosthesis that will restore full motor and sensory capability to upper-extremity amputees. The VIE was developed to support that effort by providing a standardized framework for neuroscientists, engineers, and clinicians to rapidly develop and integrate the complex systems that make up the RP2009 Modular Prosthetic Limb. To make it as effective and as broadly used as possible, standard functions and interfaces exist and efforts are under way to make the VIE system open source.

A significant benefit of a virtual collaborative environment is the ability to evaluate the command and control of neuroprosthetics within simulated rendered virtual environments. In the case of advanced upper-limb prostheses (characterized by a large number of controllable degrees of freedom), a virtual environment allows immediate access to a simulated limb system for the purpose of training and control algorithm refinement. Practically, this allowed parallel development of the control systems even while physical limb hardware was still in early development stages. The VIE also enabled evaluation of prototype limb designs with end users in a clinical environment, allowing recording and playback of control signals without safety issues.

Currently, several collaborative neuroprosthetic frameworks exist, including the brain–computer interface (BCI) system BCI2000, NeuroComm, musculoskeletal modeling software (MSMS) system, and the Thought Translation Device.[4–6] Each of these frameworks provides some level of modularity. However, none serves as a platform for end-to-end development of neuroprosthetic devices that utilizes both invasive and noninvasive recording devices, supports multiple signal analysis modules, allows hardware-in-the-loop development, and provides a clinical interface. The VIE combines features of existing development frameworks into a package optimized for collaborative development and competitive assessment of neuroprosthetic system components. These features include a real-time processing engine, modular construction of a neuroprosthetic system, and an interactive rendered 3-D environment.[7]

Presently, the VIE is used by neuroscientists, control and robotics engineers, robotic limb developers, and clinical research groups. Neuroscientists use this system to run primate closed-loop BCI experiments involving penetrating electrodes.[8] Controls and robotic engineers develop control algorithms to model prototype limb systems and virtual environments in which they simulate interactions between the limb and its surroundings.[9] Finally, clinicians use the VIE graphical user interface to tailor a prosthesis to an individual patient and tune system parameters prior to and during clinical evaluation.[3]

The goal of this article is to describe the VIE architecture and initial applications. Specifically outlined are each of the five main subsystems and use cases demonstrating noninvasive myoelectric prosthesis control, as well as neural signal decoding in a primate model teleoperating a prosthetic device.

THE VIE: FRAMEWORK AND ARCHITECTURE

The VIE uses the MATLAB and Simulink (The MathWorks, Inc., Natick, MA) tool families, which provide a means to rapidly prototype embedded software and perform hardware-in-the-loop simulation. Simulink (with Simulink Real-Time Workshop and xPC Target) provides real-time system modeling and simulation, and MATLAB provides graphical user interfaces to the system. Additionally, the VIE uses third-party software modules, including Delta3D with its Open Dynamics Engine (http://ode.org/) to supplement the system with real-time collision detection and MSMS,[4] which renders the 3-D virtual world.

The full VIE hardware configuration consists of three personal computers (PCs): an operator PC, a visualization PC, and a real-time PC. The operator PC acts as a host running MATLAB to provide graphical control of the system. The visualization PC, in conjunction with a set of 3-D goggles, provides a rendered real-time 3-D stereo display of the virtual environment. Finally, the real-time PC runs the xPC Target fixed-step real-time kernel that executes signal processing algorithms and models closed-loop control.

The key advantage of the tools selected for the VIE is that the same code prototyped for simulation can be embedded directly into real prosthetic limb systems.

The VIE consists of five core subsystems (Fig. 1): inputs, signal analysis, controls, plant, and presentation. Each subsystem can be configured modularly depending on the particular application and use-case scenario of the VIE.

Inputs to the VIE

The inputs subsystem is responsible for communicating with hardware interfacing to external data sources and packaging the received data in a standardized format called the inputs bus. This allows for modular use of different types of signal acquisition hardware for live signal capture or playback of prerecorded data. In this way, downstream signal analysis algorithms can be completely isolated from the specifics of how the data reach-

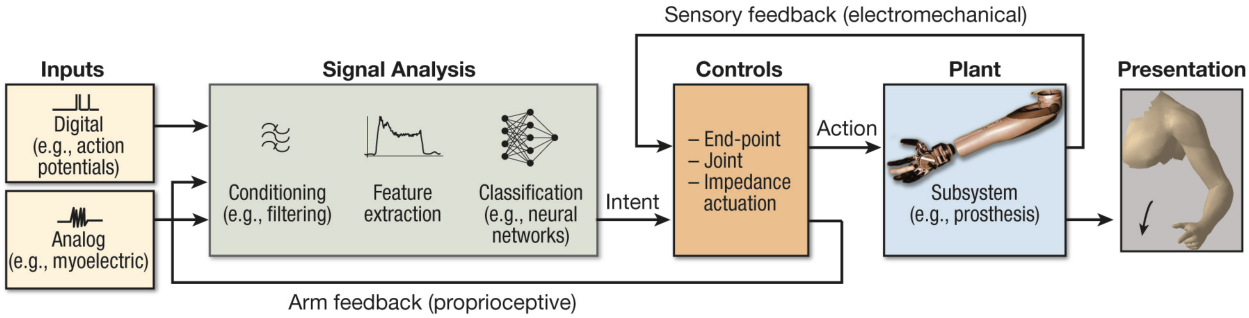


Figure 1. Block diagram of the VIE modules.

ing them have been acquired. This module allows data in various formats to conform to a standard interface. As shown in Fig. 2a, the inputs can come from a wide variety of sources, such as cortical implants, peripheral nerve implants, or intramuscular or surface electromyography electrodes, as well as mechanical devices (e.g., contact switches or joysticks), motion tracking sensors, or files containing prerecorded data.

Signal Analysis

The signal analysis subsystem processes the incoming (live or prerecorded) signals, extracts signal features, and then converts the signals into a command to the limb system, as shown in Figs. 1 and 2b. Signal feature extraction is used for pattern classification algorithms to reduce dimensionality of input signal. The particular feature set and classifier/regressor are chosen on the basis of the intended application and available signal set. Algorithm parameters (weighting matrices, etc.) associated with these algorithms are updated during a "training" phase during which processed input signals are correlated to known movements. This process of training, or correlating input signals to intended movements, is an offline process coordinated by the VIE that allows the algorithm parameters to be customized to an individual limb user.

In a myoelectric control scenario, the signal analysis subsystem performs bandpass filtering of input signals, extracts time-domain features (e.g., mean absolute value and number of zero crossings during a given time window), then uses a linear classifier to select the most likely intended movement class, and finally issues commands to the controls subsystem. Each of these processing elements can be modularly selected depending on the specific application. Examples are provided in the *Applications* section.

The output of the signal analysis block is an action command that represents the high-level intent of the end user (e.g., flex elbow, open hand, or move to location). These position and/or velocity commands are transferred using the common interface either to a controls block or directly to physical limb hardware outside of the simulation environment.

can we get to motor level

Controls

The controls subsystem of the VIE receives high-level commands representing the intent of the neuroprosthetic end user (e.g., flex elbow to 90°) and translates these commands into low-level actuator commands (e.g., supply current for motor commutation). These commands can be used for hardware-in-the-loop simulation or for high-fidelity motor simulations. Using the combination of desired commands from the end user and feedback from the modeled system, the controls block regulates low-level actuator commands to achieve the desired system performance.

The intent bus from the signal analysis subsystem contains command information (Fig. 2) including joint commands, Cartesian space commands, and movement macros (e.g., hand grasp patterns). Grasp commands use customizable trajectories for each finger of the prosthetic device, allowing high-level coordination to accomplish predefined hand conformations.

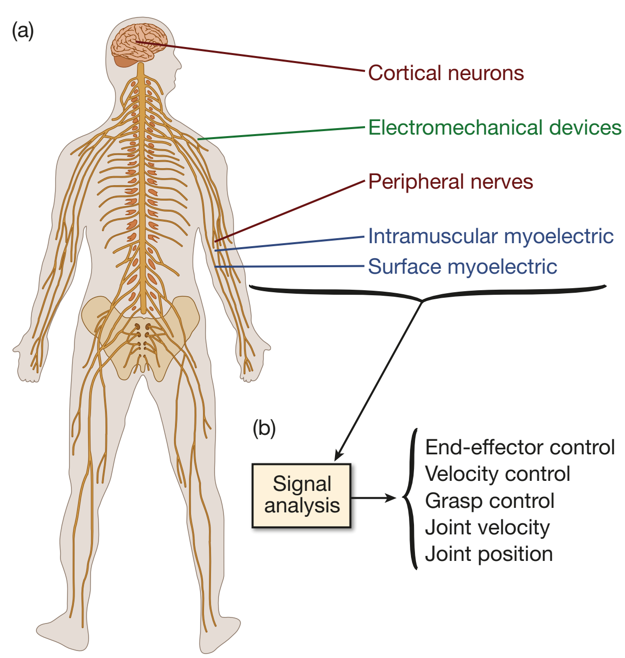


Figure 2. (a) Physiological inputs to the VIE. (b) Signal analysis inputs and outputs.

The controls system modulates the impedance or apparent stiffness of selected joints, allowing the limb to mimic the compliance of the natural limb. Modularity within the controls subsystem provides the flexibility to evaluate the efficacy of a variety of control modalities. This allows the end user to switch between a number of different control strategies depending on the task at hand or whichever mode of operation the user prefers.

The controls subsystem (Fig. 3) has three main functional blocks representing high-level controls, motor controls, and the plant (i.e., neuroprosthetic device). Each block has an input and output vector and the various controllers interact dynamically with the plant block (described in more detail in the next section). The nonlinear state space equations inside the plant block represent the prosthetic limb segments, the actuators, and the mechanical transmissions that connect them. The plant input is a vector of motor voltages and output is a vector of plant state variables representing motor currents, motor angular positions, and motor angular velocities. This plant state vector also contains link angular positions and velocities that cannot be written as algebraic functions of other state variables. The plant output is a vector of sensor measurements including current, torque, and motion of the electromechanical limb system.

The VIE control architecture serves as an extremely flexible test bed for investigating command/control paradigms for the end user of the prosthetic limb in an effort to evaluate and determine optimal function for human-in-the-loop performance.

Plant

The plant subsystem of the VIE models the underlying physical prosthetic device and components being developed. Within the plant block, each prosthetic joint is modeled to reflect its kinematic and/or dynamic behavior. The limb system follows a sequential kinematic hierarchy from proximal (shoulder segment) to distal (the hand). The proximal segment represents the attachment point of the prosthetic device and can be moved in space to reflect torso motion (using motion sensors) in the immersive environment of the VIE.

There are various selectable levels of fidelity and assumptions made within the plant model (e.g., ignoring motor back electromotive force, friction, or inertial terms) that can be customized to either increase simulation accuracy as in the highest-fidelity cases or to decrease computation time as in cases where the limb simulation should run interactively in real time. To perform accurate control system analysis, a dynamic plant model is used. In this case, the plant models the nonlinear multibody rigid dynamics of the prosthetic arm linkages along with the electromechanical actuator models that drive each link. Conversely, for more basic simulations that require only animations of limb motion, the plant can be configured as a fast kinematic model (ignoring inertial effects) with direct feedthrough of desired link/motor commands.

A distinct advantage of using a simulated plant model is that it reduces hardware limitations including safety concerns, cost, and limited availability. Additionally, the VIE system has been used to operate hardware in the loop to evaluate control algorithms and physical performance of actuators on a test stand environment.

Presentation

The presentation subsystem of the VIE bridges the simulated environment to the outside world. As shown in Fig. 4, state data of the system (consisting of joint position information of a prosthetic limb and location of interactive objects in the virtual world) are transmitted via User Datagram Protocol to a networked computer that renders the 3-D display. The end user can observe the virtual world from a third-person perspective or may use a stereoscopic display to gain a first-person perspective of the 3-D environment, in which case the end user peers through the eyes of the avatar. By providing a realistic means to visualize prototypical prosthetic devices, end users have an opportunity to work with a high-fidelity simulated prosthesis and customize or tune the device according to their ability and preference within the safety of a virtual environment.

Interaction with virtual objects between the limb and the synthetic environment occurs in three steps. First,

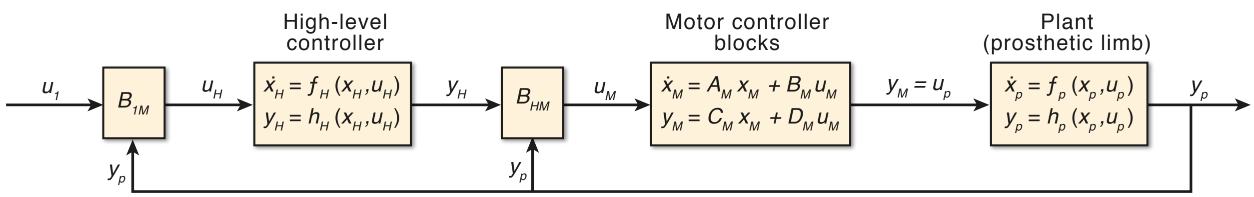


Figure 3. Block diagram of prosthetic limb control system. Inputs (u) and outputs (y) represent signal vectors between processing blocks, and B_{HM} and B_{IM} simply sort the inputs to the appropriate motor controller subsystem. The high-level controller block contains a set of nonlinear state space equations representing nonlinear endpoint control algorithms to calculate the appropriate motor commands. The motor controller blocks contain a set of linear state space equations (described by the matrices $\{A_M, B_M, C_M, D_M\}$) outputting motor control voltages for each joint. The plant input is a vector of motor voltages and a vector of state variables representing motor currents and angular positions/velocities. The plant outputs sensor measurements including current, torque, and motion of the limb.

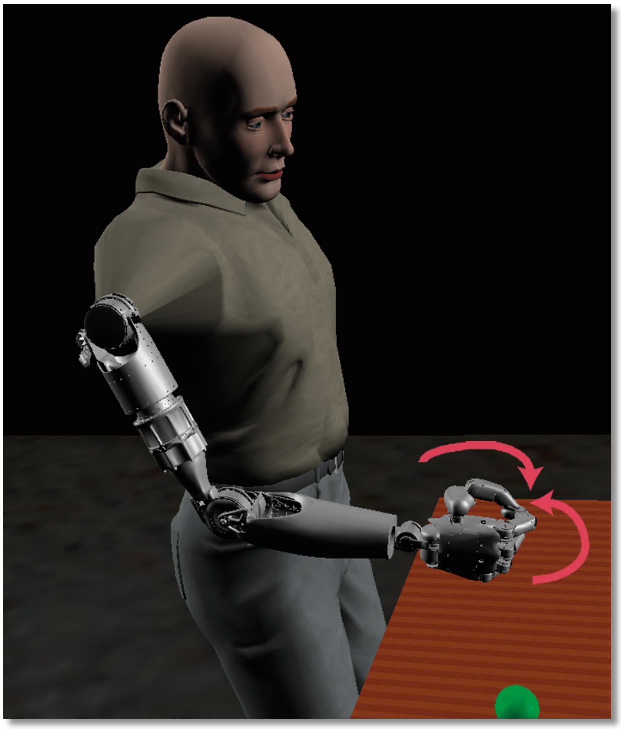


Figure 4. Presentation environment depicting a precision grip.

the plant dynamics mathematically represent the device including actuator electromechanical properties, limb segment dynamics, joint degrees of freedom, and forces and torques from the external world. Second, the physical volume of the limb is modeled as geometric primitives for collision detection between these geometric primitives and the objects of the virtual world. Finally, a rendered graphics scene is presented to the end user. Separating the problem of world interaction into these three parts distributes the computational load and allows efficient calculation of dynamic properties, fast collision detection, and high-quality scene rendering. The dynamic response of the prosthetic device is computed on the real-time PC using SimMechanics (The MathWorks), and the segment angles of the prosthesis are broadcast by the presentation block.

To perform the collision detection associated with the virtual environment, custom software based on the Open Dynamics Engine receives the segment angles and updates its own geometric-primitive prosthesis model. On the basis of the relative position of the limb and world objects, the Open Dynamics Engine-based collision detection engine updates the physics models of world objects external to the limb. These virtual interactive objects use geometric primitives with rigid first-order dynamic response. The resulting position and orientation of collided objects is then broadcast (via User Datagram Protocol) for 3-D rendered visualization. In this way, limb interaction with the virtual world appears seamless and realistic.

In a clinical setting, the presentation tools within the VIE can serve as a proving ground for the neuroprosthetic device. The prosthetist and patient can work together to tune the virtual limb system responsiveness (i.e., actuation speeds) and then later load these configuration parameter sets onto the real prosthetic device. The patient can then perform a variety of functional tasks in the virtual environment. Data from these tasks can be recorded and used to evaluate how effectively the user can control the limb (Fig. 5).

APPLICATIONS

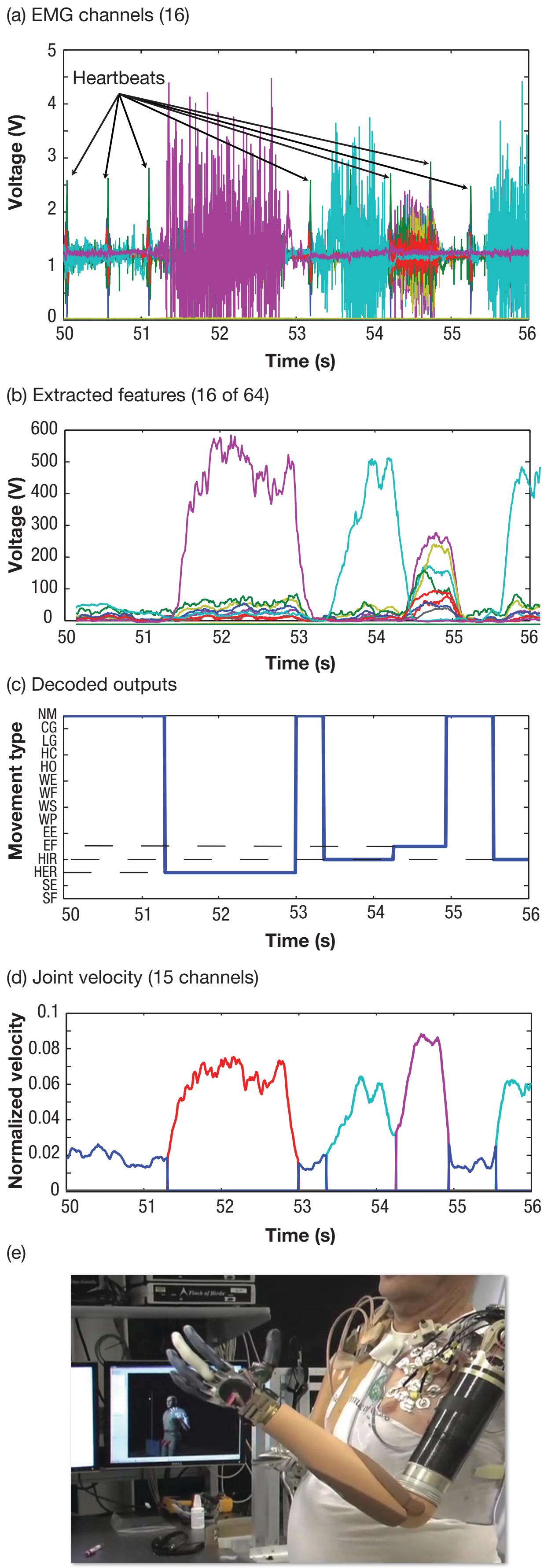
Real-Time Myoelectric Decoding

During clinical research using the VIE, noninvasive myoelectric activity is acquired from skin electrodes placed on the residual muscle of an amputee (Fig. 5a).[3,9] The signals are fed to the VIE, which processes them by bandpass filtering the signals between 80 and 500 Hz and then extracts four time-domain features from the filtered signals for each electrode (Fig. 5b). The four features are then classified into 14 movement classes (plus a "no movement" class) using linear discriminant analysis.[9] These signals, in conjunction with the joint velocity signal (Fig. 5, c and d) are then used to control a virtual arm (Fig. 5e), as well as an actual prosthetic limb developed as part of the RP2009 program. The control signals produce highly accurate movements, and execution times for these movements are comparable to those of normal subjects.[3] As shown in Fig. 5e, the subject is outfitted with surface electrodes on his chest to record upper-limb muscle activity. The movements performed in these experiments, shown in detail in Fig. 5c, were broad hand/arm movements (e.g., elbow flex/extend, wrist flex/extend, hand open/close, etc.), but it is also possible to decode flexion and extension movements of individual fingers from myoelectric activity collected from transradial amputees.[11,12] In this case, it is worth noting that commercial prostheses cannot perform individuated finger movements, making it hard to visualize output movements. However, the VIE can address this shortcoming by allowing visualization of any joint movement within the virtual environment.

Remote Neural Decoding

Because of its modularity, the VIE can perform signal processing at one location and operate hardware at remote locations. This concept was established and tested using the VIE framework to control a virtual prosthetic limb at one location via the Internet using a neural decode algorithm of a primate at a collaborator's lab at a separate location.

In this setup (Fig. 6), two VIE systems were used, one at each lab. A data acquisition system for neural signals was used to collect cortical action potentials in a non-



human primate in real time. The action potential spikes were sorted into bins and sent to the inputs block of the VIE system.

In this animal model, the primate was presented a 3-D target using shutter glasses with a stereoscopic display. Single- and multiunit activity was correlated to the position of a "cursor" that was controlled neurally to attain the target.

The target was correlated to the wrist location of the prosthetic arm, and corresponding joint angles were calculated using inverse kinematics. The endpoint command was bounded by the working volume of the prosthetic arm, and the joint angles were solved. The control algorithm computed the error between the current joint angles and the desired joint angles and computed appropriate joint actuator commands using a simple proportional gain controller.

A kinematic plant model was implemented on the VIE performing neural decoding. Using realistic angular velocity limits, the plant model joints were driven according to the actuator commands assuming zero load on the limb.

One VIE system (VIE$_1$ in Fig. 6) was responsible for decoding neural spike activity to resolve cursor position in space. The cursor location was converted to an equivalent set of joint angles for a 7-degree-of-freedom arm, such that the position of the wrist corresponded to that of the cursor. These joint angles were transmitted by the visualization block and locally repackaged by the decoding VIE using Transmission Control Protocol/Internet Protocol (TCP/IP) and broadcast over the Internet to our institution in Maryland.

The remote VIE (VIE$_2$ in Fig. 6) received the message over TCP/IP and used these joint angles to control a virtual prosthetic limb in real time. This ability to perform remote decoding and limb actuation has been tested using both primate training data sets and real prosthetic limb hardware, as well as live decoding of primate data

Figure 5. (a) Electromyogram (EMG) data acquisition from 12 skin surface locations; (b) feature extraction (one of four feature outputs shown); (c) classification: the VIE outputs a movement class; and (d) an output velocity signal for that class. These outputs are then used to drive both a real and a virtual upper-limb prosthesis (e). Note that the EMG data show a heartbeat signal with the EMG signals themselves because of the fact that the shoulder disarticulation patient, from whom the data were collected, underwent a targeted reinnervation surgery, whereby the peripheral nerves are rerouted to innervate the pectoral muscles (also see Ref. 10 for more on this procedure). CG, cylindrical grasp; EE, elbow extension; EF, elbow flexion; HC, hand close; HER, humeral external rotation; HIR, humeral internal rotation; HO, hand open; LG, lateral grasp; NM, no movement; SE, shoulder extension; SF, shoulder flexion; WE, wrist extension; WF, wrist flexion; WP, wrist pronation; WS, wrist supination.

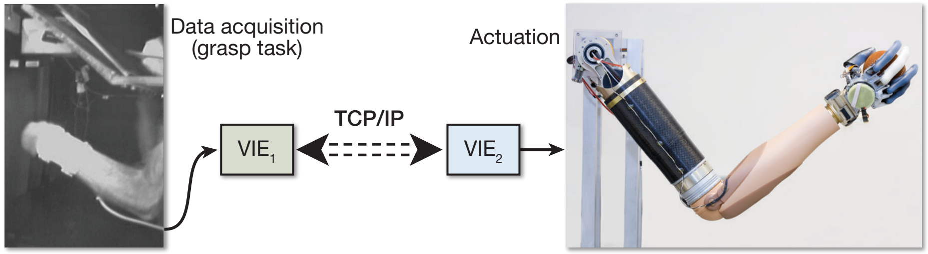


Figure 6. Neuroscientists and rehabilitation engineers can use the VIE to acquire biological signals and control a robotic arm from distinct locations.

controlling a virtual prosthetic. The modularity of the VIE allows these experimental paradigms to be configured interchangeably.

The use of multiple VIE systems in this capacity demonstrates the capability to accommodate real-time decoding from live sources and to use those data remotely to control a neuroprosthetic device in a safe and isolated environment.

Summary of Other VIE Results

The VIE has been used in numerous other applications where an actual upper-limb prostheses was not available, or where the appropriate degrees of freedom that needed to be controlled were lacking on the prostheses, or as a complement to a real prosthesis for visualization purposes. Prerecorded data are also frequently used on the VIE to test new approaches to decoding datasets.

For example, primate studies have shown that it is possible to decode individual finger and wrist flexions and extensions from neural signals in the primary motor cortex (M1) hand area.[1,8] In this context, the VIE was used to show asynchronous decoding of these movements to confirm the validity of the decoded movements.

Additionally, the modular nature of the VIE lends itself to the direct comparison of algorithms. For example, in one study,[13] a linear filter and recursive Bayesian decoder were both implemented in the VIE. The computational cost and real-time behavior of each algorithm were examined. Notably, this was the first reported real-time implementation of the Mixture of Trajectory Models decoder.[14]

DISCUSSION

Novelty and Benefits Introduced by the VIE

A major advantage of the VIE system is the modular structure, which allows varying levels of model fidelity and sophistication within each subsystem depending upon the desired use of the VIE.

This is similar to, but distinct from, other neuroprosthetic development frameworks, such as BCI2000,[5] which is composed of four modules: data acquisition and storage, signal processing, user application, and operator interface. The BCI2000 modules communicate through TCP/IP, so they do not have to be collocated, which is an advantage shared by the VIE. Although BCI2000 has been successfully used for a number of applications, it has limitations because it does not separate the physical system (i.e., the plant) from the control modules. This implies that the system does not allow a way to combine plant models of different prosthetic limbs with control algorithms. BCI2000 also lacks provisions for third-party dynamic engine software packages, making it difficult for end users to create dynamic simulations of models of a prosthetic limb. Although arbitrary "user application" modules can be deployed, BCI2000 does not natively support 3-D visualization, which is essential for realistic evaluations/simulations. Most of the shortcomings of BCI2000 are associated with modeling and controlling a physical limb as opposed to a virtual cursor. In contrast, the MSMS package was specifically designed and integrated into the VIE to facilitate modeling and dynamic simulations of neurally controlled prosthetic limbs.[15] The key features of MSMS include the ability to develop musculoskeletal or mechanical models of limbs in a standard file format, a user-friendly graphical user interface to construct and manipulate components of these models, integration with the Simulink dynamic solver, and integrated 3-D (virtual reality) graphics routines to visualize the simulations.

Limitations of the VIE

The VIE system represents many different capabilities including signal processing, hardware-in-the-loop control, and virtual reality visualization. In order to accomplish these tasks, multiple computer systems were used, making the system not readily portable. Although the VIE is scalable enough to operate on a single laptop, performance can be limited in this configuration. Typical research systems require two dedicated PCs, and clinical systems with stereo vision require three PCs. This represents a limitation in the lab space required and the expense of establishing the system. The implementation of the system natively in MATLAB also poses some difficulties in organizing the programming effort on a large scale. Although the Simulink Coder facilitates code translation to embedded software, developers are still responsible for writing efficient algorithms that meet the demands of a real-time scheduler (typically a 1-ms time step).

Applications of the VIE

The *Applications* section describes two distinct application environments for the VIE: a clinical environment and a research environment. The VIE serves as a tool for clinical evaluation of neuroprosthetics technology at all stages of the fitting process, from myography site selection to performance evaluation. The technology includes but is not limited to prosthetic limbs, sensory feedback devices, motor decoding algorithms, sensory feedback algorithms, patient training algorithms/regiments, and the use of virtual environments for limb system evaluation and fitting. The modularity of the VIE allows the system to be used in a research environment to develop and validate performance of novel decoding algorithms, as well as in a clinical environment to refine an end user's command and control of a prosthetic device. The graphical user interface allows configuration and display of parameters necessary for performance tuning. From a research perspective, the VIE provides a framework to efficiently design and develop neuroprosthetic components using standard interfaces on a common platform.

CONCLUSIONS AND FUTURE WORK

In this work we have shown the applicability of a novel integrated framework for development, simulation, modeling, and clinical evaluation of neuroprostheses. The modularity and thorough documentation of the VIE allows for successful integration of the framework in ongoing clinical evaluations and experiments at several locations around the country. Given the distributed nature of the RP2009 program (more than 30 institutions in five countries), the VIE has successfully allowed researchers and developers in distinct geographical locations to accomplish significant development efforts in a short period of time.

Although the VIE has functioned well with its current architecture, there are improvements that will enhance its versatility for researchers and clinicians alike. The VIE was originally designed with a focus on upper-extremity prostheses, however, with further development and extensions, the VIE will serve as a general development framework for any type of neural prosthetic device. These changes include generalizing existing bus structures to include a wider range of prosthetics systems, adding an experimental control block to control the flow of experiments, and adding a block for data logging.

ACKNOWLEDGMENTS: This work was funded and supported by the DARPA RP2009 program. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. government.

REFERENCES

[1]Schieber, M. H., and Hibbard, L. S., "How Somatotopic Is the Motor Cortex Hand Area?" *Science* **261**(5120), 489–492 (1993).

[2]Baker, J., Yatsenko, D., Schorsch, J., DeMichele, G., Troyk, P., et al., "Decoding Individuated Finger Flexions with Implantable Myoelectric Sensors," in *Proc. IEEE Engineering in Medicine and Biology Society Conf.*, Vancouver, BC, Canada, pp. 193–196 (2008).

[3]Kuiken, T. A., Li, G., Lock, B. A., Lipschutz, R. D., Miller, L. A., et al., "Targeted Muscle Reinnervation for Real-Time Myoelectric Control of Multifunction Artificial Arms," *J. Am. Med. Assoc.* **301**(6), 619–628 (2009).

[4]Davoodi, R., and Loeb, G., "A Software Tool for Faster Development of Complex Models of Musculoskeletal Systems and Sensorimotor Controllers in Simulink," *J. Appl. Biomech.* **18**(4), 357–365 (2002).

[5]Schalk, G., McFarland, D., Hinterberger, T., Birbaumer, N., and Wolpaw, J., "BCI2000: A General-Purpose Brain-Computer Interface (BCI) System," *IEEE Trans. Biomed. Eng.* **51**(6), 1034–1043 (2004).

[6]Hinterberger, T., Mellinger, J., and Birbaumer, N., "The Thought Translation Device: Structure of a Multimodal Brain-Computer Communication System," in *Proc. IEEE EMBS Conf. on Neural Engineering*, Capri Island, Italy, pp. 603–606 (2003).

[7]Bishop, W., Armiger, R., Burke, J., Bridges, M., Beaty, J., et al., "A Real-Time Virtual Integration Environment for the Design and Development of Neural Prosthetic Systems," in *Proc. IEEE Conf. of the Engineering in Medicine and Biology Society*, Vancouver, BC, Canada, pp. 615–619 (2008).

[8]Aggarwal, V., Acharya, S., Tenore, F., Shin, H. C., Etienne-Cummings, R., et al., "Asynchronous Decoding of Dexterous Finger Movements Using M1 Neurons," *IEEE Trans. Neural. Syst. Rehabil. Eng.* **16**(1), 3–14 (2008).

[9]Tenore, F., Armiger, R., Vogelstein, R., Wenstrand, D., Englehart, K., and Harshbarger, S., "An Embedded Controller for a 7-Degree of Freedom Prosthetic Arm," in *Proc. IEEE Engineering in Medicine and Biology Society Conf.*, Vancouver, BC, Canada, pp. 185–188 (2007).

[10]Kuiken, T. A., Dumanian, G. A., Lipschutz, R. D., Miller, L. A., and Stubblefield, K. A., "The Use of Targeted Muscle Reinnervation for Improved Myoelectric Prosthesis Control in a Bilateral Shoulder Disarticulation Amputee," *Prosthet. Orthot. Int.* **28**(3), 245–253 (2004).

[11]Tenore, F., Ramos, A., Fahmy, A., Acharya, S., Etienne-Cummings, R., and Thakor, N. V., "Decoding of Individuated Finger Movements in Transradial Amputee Using Surface Electromyography," *IEEE Trans. Biomedical Eng.* **56**(5), 1427–1434 (2009).

[12]Armiger, R., and Vogelstein, R., "Air-Guitar Hero: A Real-Time Video Game Interface for Training and Evaluation of Dexterous Upper-Extremity Neuroprosthetic Control Algorithms," in *Proc. IEEE Biomedical Circuits and Systems*, Baltimore, MD, pp. 121–124 (2008).

[13]Bishop, W., Yu, B., Santhanam, G., Afshar, A., Ryu, S., et al., "The Use of a Virtual Integration Environment for the Real-Time Implementation of Neural Decode Algorithms," in *Proc. IEEE Engineering in Medicine and Biology Society Conf.*, Vancouver, BC, Canada, pp. 628–633 (2008).

[14]Yu, B. M., Kemere, C., Santhanam, G., Afshar, A., Ryu, S. I., et al., "Mixture of Trajectory Models for Neural Decoding of Goal-Directed Movements," *J. Neurophysiol.* **97**(5), 3763–3780 (2007).

[15]Hauschild, M., Davoodi, R., and Loeb, G., "A Virtual Reality Environment for Designing and Fitting Neural Prosthetic Limbs," *IEEE Trans. Neural Systems and Rehabilitation Engineering* **15**(1), 9–15 (2007).

The Authors

Robert S. Armiger

Francesco V. Tenore

William E. Bishop

James D. Beaty

Michael M. Bridges

James M. Burck

R. Jacob Vogelstein

Stuart D. Harshbarger

Robert S. Armiger is a Section Supervisor for the Biomechanics Section of the Biomedical Engineering Group and was a lead developer of the VIE, implementing the design and participating in clinical evaluation of the VIE and prosthetic limb systems for the Revolutionizing Prosthetics (RP) program. **Francesco V. Tenore** is an Assistant Section Supervisor and currently leads the Sensory Stimulation team for Phase 3 of the RP program. He contributed to the algorithm development and analysis of the electromyography signal analysis. **William E. Bishop** is currently pursuing his doctorate at Carnegie Mellon University and was a lead developer of the neural decoding components of the VIE. **James D. Beaty** is Assistant Group Supervisor for the Biomedical Engineering Group and is team lead for the Neural Decoding Experiments, coordinating research activities for both APL and partner institutions. **Michael M. Bridges** was the team lead for controls, developing and implementing the controls architecture for the VIE and prosthetic limb systems. **James M. Burck** was the team lead for the VIE and Systems Engineer for the RP program, defining system architecture and interfaces. **R. Jacob Vogelstein** was the Lead Engineer for the RP2009 Neural Interface team, which was tasked with developing implantable neural devices and architecting a common algorithm framework for the VIE and hardware systems. **Stuart D. Harshbarger** was the program manager and system architect for Phases 1 and 2 of the RP program, coordinating research and development activities across APL and numerous external partners. For further information on the work reported here, contact Robert Armiger. His e-mail address is robert.armiger@jhuapl.edu.

The *Johns Hopkins APL Technical Digest* can be accessed electronically at **www.jhuapl.edu/techdigest**.