

# Control System Architecture for the Modular Prosthetic Limb

Michael M. Bridges, Matthew P. Para,  
and Michael J. Mashner

*The Defense Advanced Research Projects Agency Revolutionizing Prosthetics 2009 program tasked APL with developing a neurally controlled prosthesis to restore function and dexterity to soldiers with upper-arm amputations. The result of this program is the Modular Prosthetic Limb (MPL), which is currently undergoing testing at the University of Pittsburgh and the California Institute of Technology with humans and nonhuman primates. In this article, we provide an overview of the human-machine interface (HMI) between the MPL and patients and discuss how the inherent flexibility of the MPL's control architecture is able to support varying HMI paradigms.*

## INTRODUCTION

The Revolutionizing Prosthetics 2009 program was created by the Defense Advanced Research Projects Agency (DARPA) with the goal of providing soldiers with upper-arm amputations with the ability to resume normal activities of daily living. Through this program, APL has developed a 17-degree-of-freedom limb known as the Modular Prosthetic Limb (MPL).<sup>1,2</sup> The design requirements for the MPL are to achieve performance as close as possible to that of a natural limb in a variety of metrics: dexterity, strength, form factor, weight, tactile feedback, and others. The design also needed to support a range of patients, from those who have lost only their hand to those with full shoulder-disarticulation injuries. The control systems developed for the MPL have been designed to allow flexibility in controlling the limb for patients, the engineers developing the limb, and researchers who may be using the limb

for other purposes. This article provides an overview of the control architecture implemented and the ways that it interfaces with users of the system.

## CONTROL SYSTEM HIGH-LEVEL ARCHITECTURE AND INTERFACES

The controls problem for the Revolutionizing Prosthetics program is akin to designing a very sophisticated human supervisory control system. Common systems with which one may be familiar include a person driving a car, flying an airplane, operating a backhoe, or playing a video game. The main components in such control systems include the machine being controlled, the environment that the machine interacts with, the human-machine interface (HMI), and, of course, the

human doing the controlling. The information flow between these components can be described as command/control signals flowing from the human to the machine, feedback information flowing from the machine and environment back to the human, and physical interactions between the machine and the environment, as shown in Fig. 1. The control objective is to allow the human controller to force the machine to do what is desired subject to the constraints of the machine, interfaces, and environment. Whether one controls a car, a video game, a backhoe, or, in our case, an advanced prosthetic limb, the above construct applies. The main differences have to do with the machine and the interface. In particular, the HMI determines the number and nature of the control “knobs” used to command the machine. Similarly, the HMI determines the nature of some of the sensory feedback information. In the case of a car, the driver has access to three control knobs: the steering wheel, the gas pedal, and the brake. Feedback information is provided by gauges on the dashboard as well as other sources; for example, the steering wheel provides feedback in the form of a restoring torque the driver feels at the wheel when making a sharp turn. The driver also experiences visual feedback from the view of the road and car; this feedback is independent of the HMI.

A backhoe operator typically manipulates six control knobs in the form of four levers to control four joints on the backhoe and two pedals to control the swing direction of the cab/shovel. The feedback is from hydraulic pressure felt through the levers, and visual feedback based on what the operator sees.

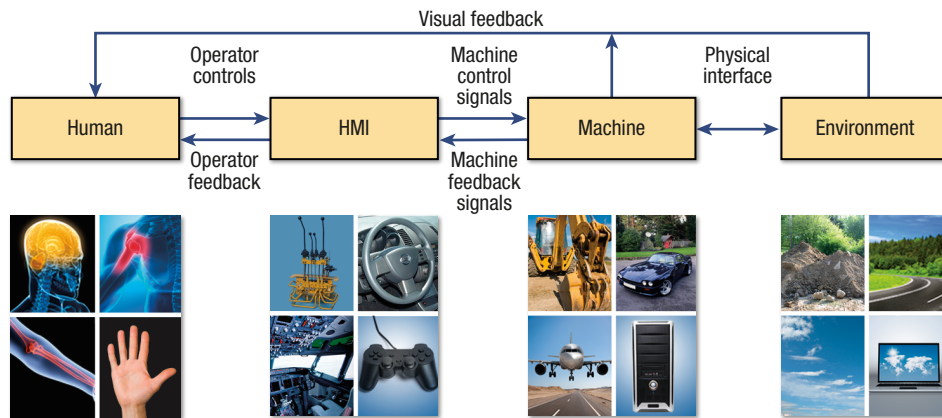


Figure 1. Depiction of information flow within a generic HMI.

and associated control algorithms. In the feedforward path, the HMI is a collection of sensors and signal processing algorithms that measure and interpret physical patient control signals (muscle contractions, brain waves, peripheral nerve implant signals) from the human and transform them into electrical machine control signals. The machine takes electrical input signals and transforms them into some kind of action (torque, motion) and drives the motors of the MPL to realize the intent of the patient. In the feedback path, the machine sensors encode information (strain, current, position) into electrical machine feedback signals that are sent back to the HMI. The HMI then decodes/interprets the electrical machine feedback signals and transforms them into a physical patient feedback signal (force, vibration, temperature, direct nerve stimulation) that is detectable by the human.

The MPL was designed to work with a wide variety of HMIs including electromyographic sensors on existing muscle sites<sup>3</sup> where, for example, reading muscle activity in the forearm of a wrist amputee provides control information regarding how the fingers should be commanded. Another HMI example is a peripheral nerve interface where an implant into the remaining periph-

## MPL CONTROLS PARADIGM

In the case of controlling the MPL, the general human-in-the-loop control system can be redrawn as shown in Fig. 2.

In Fig. 2, the machine is the actual MPL, which consists of the limb’s physical structure, motors, sensors, embedded processors,

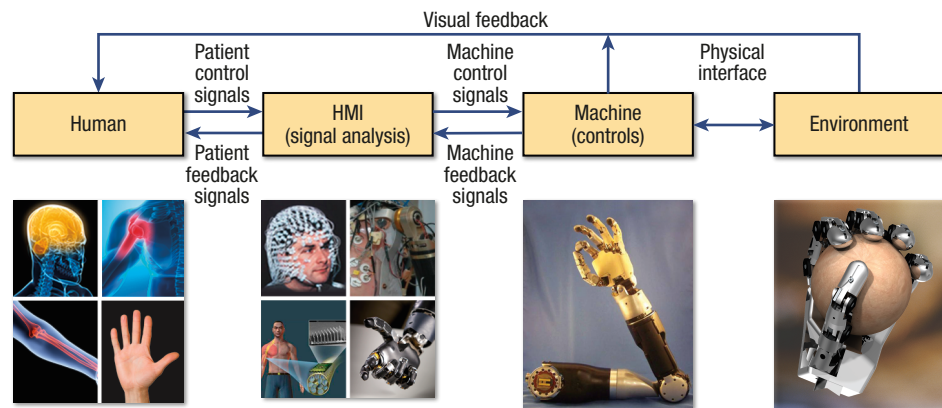


Figure 2. Depiction of the information flow between patients and the MPL.

eral nerve of a shoulder amputee is used to ascertain how upper-arm joints should be commanded. Finally, cortical implants and electroencephalography brain caps provide additional means of deciphering human intent and constructing associated commands to the machine (i.e., the MPL). In all of the above examples, the number of control knobs is a function of the number of patient signals available for decoding and the number of distinct machine command signals that can be constructed.

Depending on the HMI, there can be a number of mechanisms that provide feedback to the patient. There will almost always be a visual feedback mechanism independent of the HMI. The fact that the MPL is physically attached to the patient by means of a socket implies there is always a natural feedback path of forces and moments that the patient feels at that interface. There is also an artificially constructed feedback mechanism where, for instance, temperature and forces at the fingertips are fed back to the patient in form of direct cortical or peripheral nerve stimulation. In the absence of such an interface, a tactile actuator device called a “tactor” can be used to provide indirect feedback. Here, forces and temperatures measured by sensors at various places on the MPL hand are transmitted onto the surface of the skin somewhere on the body (e.g., the chest). The tactor contains tiny motors that press on the chest in proportion to forces measured at the fingertips. Similarly, the tactor may contain a Peltier device that can produce hot and cold sensations on the chest in proportion to what is measured at the MPL fingertips.

## MPL CONTROL SYSTEM MODALITIES

The MPL control system modalities have been designed to maximize the flexibility of the interface and the control modalities available to the patient. Given the wide variety of patients that might use the MPL, it was recognized that patients might have different control modality preferences. Consequently, the control modalities designed for the MPL consist of:

- Reduced Order Control (ROC)
- Cartesian Space Control
- Joint Space Control
- Muscle Space Control

The prosthetist works with the patient to determine what signals the patient has available to be decoded and what control modality feels most intuitive to the patient. Additional control requirements focus on smooth, natural motion and the ability of the arm to mimic the joint impedance (stiffness, inertia, and damping) of a natural arm.<sup>4</sup> The importance of joint impedance modulation was driven by the fact that interactions between the MPL and other people or objects should not cause damage to either. Each of the above modalities share a

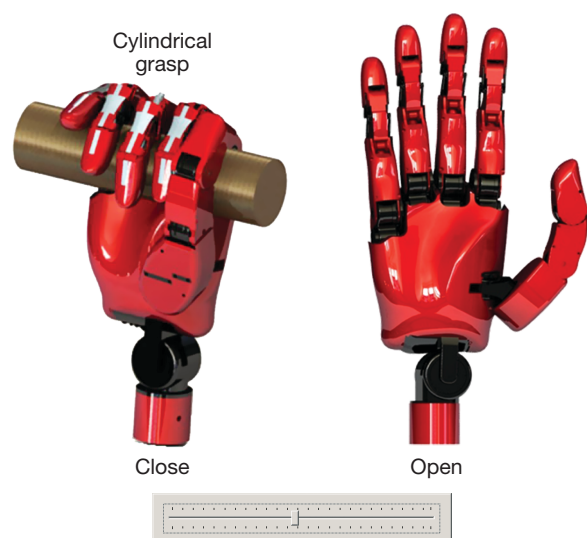
common set of underlying low-level control algorithms that can be controlled in position, velocity, and torque mode. Furthermore, while in each of these modes, the impedance can be chosen to be modulated or not. Ultimately, these control modalities describe the machine control signals issued from the HMI hardware and received by the MPL.

### ROC Example

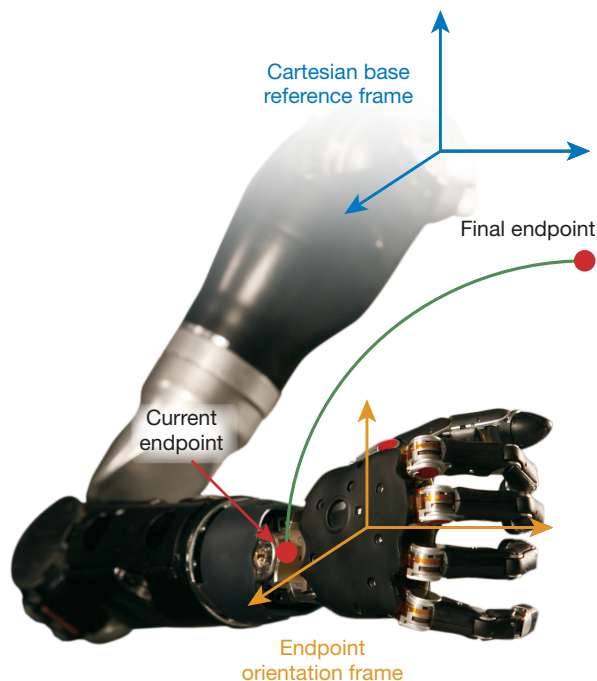
When the number of available patient control signals is less than the number of degrees of freedom, the HMI must interpret a reduced number of patient control signals and map them to a full set of machine control signals to drive individual motors. For controlling the hand, which has 27 individual joints, this ROC mode is most advantageous because most patients have less than 27 available control knobs. Preprogrammed algebraic mappings between a single machine control signal and the joints in the hand can be stored in a database and given a grasp name. In the example of a cylindrical grasp shown in Fig. 3, one machine control signal (denoted by the notional slider in the figure) represents a normalized command of how open or closed the fingers should be while being constrained in a manner to grasp a cylindrically shaped object. So, at a minimum, only two machine control signals are needed to control a highly articulated hand, one for the type of grasp and one to specify the amount of contraction.

### Cartesian Space Control Example

Research has shown that signals in the motor cortex portion of the brain can be correlated with the Cartesian space position or velocity of a point on the human



**Figure 3.** Example of one type of single ROC knob-coordinated grasp shaping.



**Figure 4.** Coordinate frames used by the endpoint control algorithm.

limb. For this mode of operation, command signals correspond to a desired Cartesian space position or the velocity of a reference point on the palm of the hand as shown in Fig. 4. Patients with motor cortex implants are more likely to use this mode and are able to control both the linear translation of the endpoint and its orientation relative to the patient. The control algorithms in the machine take the desired endpoint commands and determine how to drive each joint’s motor by computing the inverse kinematics of the MPL.<sup>5</sup>

### Joint Space Control Example

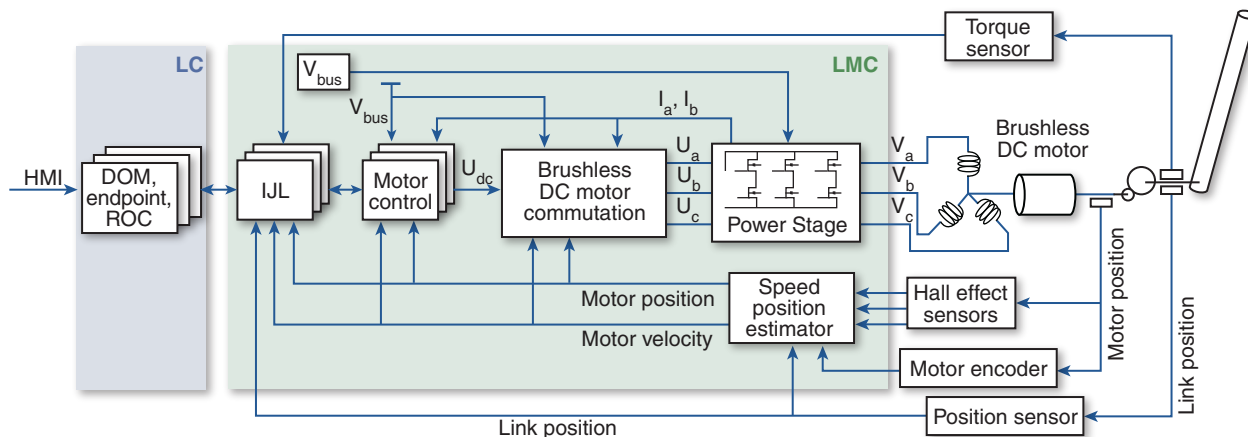
This control mode allows command signals to directly move individual joints, and in this mode, the units of the command signal can be a desired joint position, velocity, or torque. For the upper arm joints in the MPL, there is a single motor/degree of motion (DOM) for each joint. In the fingers of the hand, however, there are mechanical linkages that connect multiple joints to a single motor. Consequently, a single joint/DOM would be commanded and the remaining connected joints would be determined by the dynamics of the system.

### Muscle Space Control Example

The patient may have sensors that provide peripheral nerve signals to the MPL. These contraction signals represent force commands because they are sent to antagonistic muscle pairs around a joint. The control algorithm takes these machine command signals and turns them into motor torque commands where the level of cocontraction is used to modulate joint stiffness. One of the benefits of this mode is that it takes advantage of the natural command architecture of the human peripheral nerve system and thereby provides a more intuitive HMI.

## MACHINE CONTROL SYSTEM ARCHITECTURE AND ALGORITHMS

Whereas the previous section described the manner in which the MPL can be commanded, this section describes how the control modalities are achieved using the underlying control system architecture and algorithms. Fundamentally, the control system of the machine must take commands from the patient via the HMI and control motors. The architecture is hierarchical, and the algorithms are spread across multiple processors. Figure 5 shows, for the upper arm, how the



**Figure 5.** Machine control architecture and data flow. IJL, Individual Joint/Link; LC, Limb Controller; LMC, Large Motor Controller.  $U_a$ ,  $U_b$ ,  $U_c$ , and  $U_{dc}$  denote pulse-width modulation commands.

various control loops are nested and which sensors are used for each loop.

### Commutation and Current/Voltage Control Loop

The innermost control loop resides on the LMC processor and handles the high-rate electrical dynamics, brushless DC (BLDC) motor commutation (i.e., magnetic field switching), and current/voltage control. The motor position sensors used are for commutation and have the option of also being used in the higher-level motor control loop.

### Motor Control Loop

The next highest loop, moving from the inside out, controls the mechanical dynamics of the motor. The topology of this motor controller takes the form of a proportional integral derivative control algorithm that regulates both the position and the velocity of the motor so that the desired trajectories are followed.

### Individual Joint/Link Control Loop

Borrowing some terminology from the robotics community, links represent the solid segments of a serial manipulator while a joint represents the axis of rotation between two links. For example, in the MPL case, the humerus and the forearm are the two links adjacent to the elbow joint. Consequently, this control loop is called the Individual Joint/Link (IJL) controller because its main purpose is to control the individual position and velocity of the angle between links and the torque about a joint. The structure of this controller takes the form of an impedance control algorithm, which, when wrapped around the motor control loop, can be configured to modulate the joint stiffness, damping, and link inertia (i.e., impedance) while simultaneously controlling link motion. Alternatively, the link position and velocity can be controlled directly without modulating the impedance.

### Impedance Control

The importance of modulating joint impedance<sup>6–8</sup> flows from the requirement for the MPL to be as natural as possible. The standard industrial robot motion control problem typically places the highest importance on precise trajectory tracking and consequently creates an extremely stiff system. In the case of the MPL, a balance between tracking and stiffness must be achieved because of the interactions that occur between the limb, the patient, other people, and, in general, the outside environment. An extremely stiff system can potentially cause damage to both the patient and the MPL if a sudden impact with the environment occurs. The impedance portion of the IJL loop takes feedback from

a joint torque sensor and the link position sensor and alters the closed-loop dynamics of the system so that it more closely resembles that of a human limb. Under impedance control, when an external torque/force is applied to the MPL, the metal gears and motors are made to react with compliance similar to that of muscles and tendons. In addition, the MPL links have large moments of inertia about each joint axis because of the high gear ratios resulting from the mechanical design process. Impedance control dynamically alters the inertia to be closer to that of flesh and bone. Finally, the damping in an MPL joint can be similarly specified such that one is not stuck with the nominal damping inherent in the mechanical design.

### High-Level Controls

The algorithms mentioned so far reside on the LMC. The outermost control loop, however, resides on the Limb Controller (LC), where information from all other joints is available. In particular, the Endpoint Control (Cartesian) algorithm must coordinate the motion of multiple joints/links in order to obtain the desired Cartesian space motion. Similarly, the ROC algorithms exist on the LC and specify commands to the IJL controllers that exist on the motor controllers.

## RAPID CONTROL ALGORITHM DEVELOPMENT

After realizing the control modalities and architecture, the control implementation was designed such that control algorithms could be rapidly developed and embedded in the main microprocessor of the MPL. Referring to Fig. 5, the IJL control algorithm and the motor control algorithms are designed in Simulink and then turned into C code by the MathWorks Real Time Workshop toolbox. Once this code has been produced, it is downloaded to the LC and then integrated with the embedded code specially written by APL software engineers to control the timing, communications, and other low-level processes.

The result is an extremely flexible design that, from a controls perspective, allows:

- Patients with varying numbers and types of patient control signals to command the MPL
- Patient and prosthetists to work together to pick the control modality configurations that are best suited for the patient's needs
- The platform to be used as a test bed for algorithm evaluation
- Future lower-level control algorithm design changes and upgrades with shortened cycle time from algorithm design to implementation

## SUMMARY

The goal of this article has been to describe the controls architecture used in the MPL. Additionally, it describes the common modalities for interfacing with patients and other users of the MPL. The program is currently in a third phase where the MPL is being tested with primates, and there is ongoing work to support the continued testing and development of the limb system. Future work will include expanding upon and further developing the algorithms.

**ACKNOWLEDGMENTS:** This work would not have been possible without the contributions of the following present and past Revolutionizing Prosthetics 2009 control team members: Pat Blevins, David Carrelli, Michelle Chen, David Erickson, Eric Faulring, Jeffrey McDonald, and Emily Tai. This work was sponsored at APL under the DARPA Revolutionizing Prosthetics program, Contract N66001-06-G-8005. The views, opinions, and/or findings contained in this article are those of the authors and should not be interpreted as representing the official

views or policies, either expressed or implied, of DARPA or the DoD.

## REFERENCES

- <sup>1</sup>User jhuapl, "Modular Prosthetic Limb," *YouTube*, <http://www.youtube.com/watch?v=DjzA9b9T3d8> (uploaded 7 Mar 2011).
- <sup>2</sup>Burck, J., Zeher, M. J., Armiger, R., and Beaty, J. D., "Developing the World's Most Advanced Prosthetic Arm Using Model-Based Design," *The MathWorks News & Notes*, [http://www.mathworks.com/company/newsletters/news\\_notes/2009/jhu-model-based-design.html](http://www.mathworks.com/company/newsletters/news_notes/2009/jhu-model-based-design.html) (2009) (accessed 11 Nov 2011).
- <sup>3</sup>Lake, C., and Miguez, J. M., "Evolution of Microprocessor Based Control Systems in Upper Extremity Prosthetics," *Technol. Disabil.* **15**(2), 63–71 (2003).
- <sup>4</sup>Sensinger, J. W., and Weir, R. F., "User-Modulated Impedance Control of a Prosthetic Elbow in Unconstrained, Perturbed Motion," *IEEE Trans. Biomed. Eng.* **55**(3), 1043–1055 (2008).
- <sup>5</sup>Wedeward, K., Colbaugh, R., and Engelman, A., "Singularity Robustness: Methods for Joint-Space and Task-Space Control," in *Proc. of the 1997 IEEE International Conf. on Control Applications*, Hartford, CT, pp. 22–27 (1997).
- <sup>6</sup>Hogan, N., "Impedance Control: An Approach to Manipulation: Part I—Theory," *J. Dyn. Syst. Meas. Control* **107**(1), 1–7 (1985).
- <sup>7</sup>Hogan, N., "Impedance Control: An Approach to Manipulation: Part II—Implementation," *J. Dyn. Syst. Meas. Control* **107**(1), 8–16 (1985).
- <sup>8</sup>Hogan, N., "Impedance Control: An Approach to Manipulation: Part III—Applications," *J. Dyn. Syst. Meas. Control* **107**(1), 17–24 (1985).

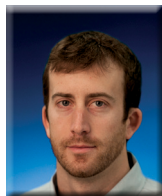
# The Authors



Michael M. Bridges



Matthew P. Para



Michael J. Mashner

responsible for control systems development, support software, and hardware integration and testing. For the current phase of the program, he is also acting as the Controls Lead. **Michael J. Mashner** is a mechanical engineer in the Air and Missile Defense Department and has been responsible for impedance control algorithm implementation and testing while also heavily influencing the overall simulation architecture for the system. For further information on the work reported here, contact Michael Bridges. His e-mail address is [michael.bridges@jhuapl.edu](mailto:michael.bridges@jhuapl.edu).

**Michael M. Bridges** is an electrical engineer in the Air and Missile Defense Department and has been responsible for the overall control architecture design along with the lower-level brushless DC motor commutation design choices and current/power limiting approaches. **Matthew P. Para** is a mechanical engineer in APL's Research & Exploratory Development Department. On the Revolutionizing Prosthetics program, he has been respon-