Modular Robot Arm Design for Physical Human-Robot Interaction

Ty Tremblay and Taşkın Padır, *Member, IEEE*
Robotics Engineering Program, Worcester Polytechnic Institute
{ttremblay,tpadir}@wpi.edu

***Abstract*—This paper describes the design and implementation of the controls and power plant for a robotic arm for physical human-robot interaction on a cyber-physical wheelchair system. There are almost 50 million people in the US who have some degree of disability, and more than 6.5 million of them experience problems with self-care. The aim of this research is to develop a system to control a modular cable-driven arm which will allow locked-in individuals, who are unable to interact with the physical world through movement and speech, to perform activities of daily living (ADL). We present the design of a compact power plant for the 5DOF arm. Modeling and control is implemented through the use of MATLAB and Robot Operating System.**

***Index Terms*—Human-in-the-Loop, Cyber Physical Systems, Physical Human Robot Interaction**

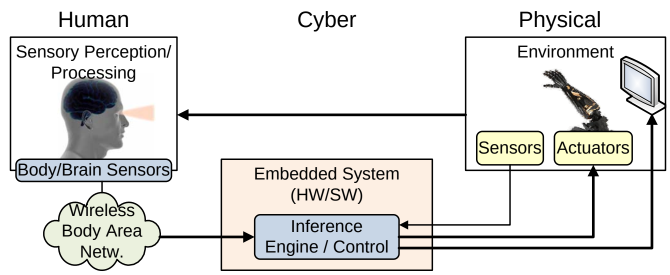

Fig. 1. A human-in-the-loop cyber-physical system.

I. Introduction

Cyber-physical systems research is finding applications in a wide spectrum of domains ranging from healthcare to transportation to energy. Reliability, safety, security and usability remain to be highest priorities in the design and implementation of CPS in which computation and communication components are tightly integrated with the dynamics of the physical components. As the CPS community continues to develop the foundations of this very challenging field, it is becoming apparent that another key component in many cyber-physical systems is the human factor [1]. For example, a recent NIST workshop report [2] identifies networked, cooperating, human-interactive CPS as a foundational research direction. As a result, human-in-the-loop cyber-physical systems (HiLCPS) emerge as a new multidisciplinary research frontier [3].

A. Human-in-the-Loop Cyber Physical Systems

Figure 1 introduces an example HiLCPS where human cognitive activity is measured through body and brain sensors; the intent is inferred through analysis on an embedded system; then human intent is translated into high-level robot control commands; robot performs the task under uncertain aim in the physical environment, where the effects are then observed by the human as an input for new decisions – closing the loop. Examples of HiLCPS, range from Brain Computer Interface (BCI) controlled robots (e.g. wheelchair) [4], to transportation systems.

This paper presents a the design of a wheelchair-mountable modular robotic arm for physical human-robot interaction within the framework of a BCI-enabled HiLCPS for persons who are functionally locked-in due to a variety of neurological or physical conditions. Locked-in syndrome is a rare but devastating condition in which an individual has full cognitive abilities but all voluntary muscles of the body are paralyzed. As a result, the individual is incapable of interacting with the physical world through movement and speech, making independent activities of daily living difficult. As of now, there is no commercially available product that could provide these individuals with control over their mobility and manipulation of surrounding objects.

In the past two decades, robotics has found applications as an assistive technology in areas ranging from autism to eldercare to stroke rehabilitation [5], [6]. Most of the initial effort has been on developing robotic technologies for intelligent wheelchairs, assistive robotic arm, intelligent prosthetics and robot assisted therapy [7], [8] . With the decreasing cost of robot components (such as sensors and actuators) and advances in artificial intelligence, the field in terms of the technology is quite mature. However, there is still a tremendous need to take the systems developed in the research labs and turn them into practical applications on the field [5].

One essential application area is the development of intelligent wheelchairs and safe robotic arms to assist physically locked-in persons [9], [10], [11], [12], [13]. The current state-of-the-art in the field demonstrates wheelchair-arm systems capable of obstacle avoidance, simultaneous localization and mapping, path planning and motion control with a shared autonomy framework. Among the important research questions for implementing a shared control of an intelligent system are: “Who takes the control of the system, human or machine, and when?”, “Under what circumstances, a decision is overridden by the humanor by the machine?”, and “How does the human-in-the-loop CPS decide on the level of autonomy in an adaptive way?”. Early efforts in the development of smart wheelchairs tackle the issue by providing the user with an external switch or button to trigger a change in the mode of operation [9], [13]. Another approach is to implement the mode change automatically where the shared control switches from human control to machine control and vice versa [14], [15].

Robotic manipulators are being developed to assist locked-in individuals, and others who need assistance, in performing daily tasks. One example is the Jaco Arm from Kinova. The Jaco Arm is a 7-DOF manipulator designed to be installed on wheelchairs to help users with daily manipulation tasks. While the arm is well suited for this purpose, its cost can be prohibitive for users. Due to its high cost, many medical assistance programs will not cover the cost of a Jaco Arm. Thus, there is a need for an inexpensive, safe, and robust robotic manipulator that is capable of performing daily manipulation tasks.

II. System Design

A. Specifications

The goal of this project was to develop a cost-effective yet capable, modular robotic arm for use in a human-in-the-loop cyber physical system. The arm can be mounted on a semi-autonomous wheelchair and is capable of assisting the user in activities of daily living. To eliminate development costs, commercially available robotic manipulators can be used. The igus robolink (Figure 2) is a cable-driven robotic arm consisting of modular joints connected via aluminum tubes. Because each joint is driven by a motor from outside the arm, many identical joints can be used in series. This reduces the cost of the arm because the number of unique parts and joints is reduced. A 5-DOF arm design can potentially meet the workspace and manipulability requirements as the two additional degrees of freedom provided by the wheelchair platform allow the arm to perform higher degree of freedom tasks.

Several specifications for the system were developed:

- The link lengths of the arm will closely resemble the link lengths of a human arm.
- The system should be able to be mounted on any electric wheelchair.
- The arm must be able to reach the floor when mounted on a wheelchair.
- The system must be capable of lifting a 1 kilogram payload.
- The arm joints must be capable of moving at 1 radian per second.
- Maintenance during the lifetime of the arm must be minimized.
- The system must cost less than $5000

B. Wheelchair

A semi-autonomous wheelchair is developed in conjunction with the arm. Wheelchair Add-on Modules (WAMs) [16] are developed as platform-independent sensor banks that can be mounted on any powered wheelchair. These modules facilitate development and application of semi-autonomous functionalities such as wall-following and door-crossing. By using the WAMs, a team of three developers can convert similar powered wheelchairs into a semi-autonomous mobility platform in less than ninety minutes. Figure 3 shows a wheelchair outfitted with the WAMs [16].

Fig. 2. An igus robolink arm.

Fig. 3. The cyber physical systems wheelchair.

C. Arm

The manipulator arm selected for the CPS wheelchair is a commercially available manipulator from Igus, Inc. called the robolink (Figure 2). The robolink is a cable driven serial manipulator with five degrees of freedom. The joints are driven via cables that are routed through the arm allowing the motors to be housed within the wheelchair. Removing the motors from the arm reduces weight and increases user safety.

The robolink configuration presented here consists of five revolute joints. The base joint (*joint0*) is a swivel joint and rotates about its z-axis which is oriented along the first link of the arm. The next joint is a compound joint combining a rotation about a z-axis perpendicular to the first link of the arm (*joint1*), a rotating joint, and a rotation about a z-axis along the second link of the arm (*joint2*). The final joint is similar to the previous joint and contains *joint3* and *joint4*. Table I shows the DH-table of the configuration of the arm as chosen for its implementation on the CPS wheelchair.

TABLE I

TABLE II-B: THE DENAVIT HARTENBERG TABLE FOR THE ARM.

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	$\frac{\pi}{2}+q_0$
2	0	$-\frac{\pi}{2}$	l_1	0
3	0	$\frac{\pi}{2}$	0	q_1
4	0	$-\frac{\pi}{2}$	l_2	q_2
5	0	$\frac{\pi}{2}$	0	q_3
6	0	0	l_3	q_4

Each joint contains a magnetic incremental encoder and a hall effect sensor. The encoders have a resolution of 0.073 degrees per tick for swivel joints and 0.078 degrees per tick for rotating joints. The hall effect sensor is used to home the joints to their zero position. Cables for the encoder and hall effect sensor are routed through each joint, out the base of the arm, and terminate in standard DB-15 connectors.

D. Drive Module

A custom drive module was developed to drive the joints of the arm. The drive module houses the five motors for the arm, six motor controllers, and all of the conectors needed to communicate with the computer and arm.

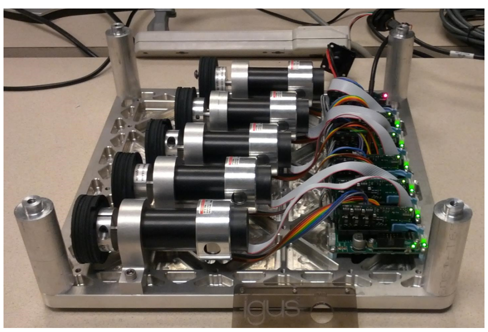

Fig. 4. The drive module for the WPI Robolink

The activity of daily living (ADL) that the arm will most commonly perform is the mainpulation of common household objects. Through this use case, the arm must be capable of lifting a 1kg weight as well as moving each joint at 1 radian per second. In order to accomplish these requirements, Maxon EC-45 Flat 50 Watt motors were selected. Each motor is capable of producing 82.7mNm of torque and has a maximum rotational speed of 3000rpm. Each motor is attached to a planetary gearbox with a 318:1 reduction resulting in a maximum output torque of 6Nm and a maximum speed of 1 radians per second. The maximum torque is limited by the torque that the planetary gearbox is cabaple of transmitting. Attached to the motors are 512cpr quadrature encoders resulting in a resolution of .002 degrees per tick after the planetary gearbox.

To control the motors, EPOS2 36/2 controllers were selected. The controllers were chosen for their small form factor and modularity. Each EPOS2 36/2 consists of the controller and a motherboard that breaks controller IO out to standard connectors. The motherboard design allows for both power and communication to be chained from controller to controller, reducing the number of cables in the drive module, increasing ease of assembly, and increasing modularity. Analog and digital IO are supported on the controller, and can be used in the future to further increase the drive module's capability. The controller motherboards have an auxiliary port that can be used, through the addition of breakout boards, for an additional encoder, communication, or additional IO. Encoder breakout boards were selected to take advantage of the encoders in the joints of the robolink. The controllers support absolute and relative position control as well as velocity control.

The drive module was designed to be as reconfigurable as possible. The module's dual plate design allows it to be mounted to the wheelchair in almost any direction. A two inch square hole pattern was machined into the module's plates to allow for easy mounting of additional devices or sensors. The module's top plate is removable with four screws, increasing ease of repair and replacement of components. The arm mount fits into a slot in each plate, allowing the arm to be removed from the module without the removal of additional hardware. Each motor is face mounted to a bracket that can be easily bolted into the drive module using two screws. Similarly, the motor controllers plug into motherboards that are fixed to the drive module using standoffs and screws. The motor controllers implement the CAN communication protocol, allowing the controllers to be daisy-chained. Adding and removing controllers does not change the number of communication ports needed on the drive module, increasing modularity and reconfigurability.

E. Software Design

Control software for the arm was developed using ROS and written in Python and C++. Figure 5 shows a basic diagram of the software architecture.

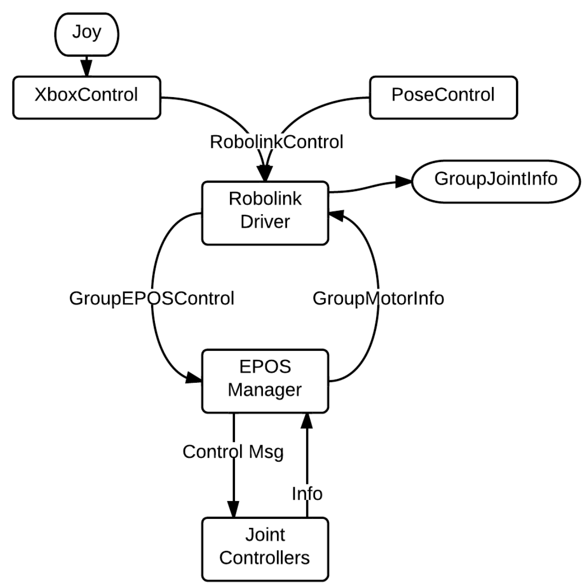

Fig. 5. A diagram of the control software for the arm.

Four ROS nodes were created to control the arm. The two highest level nodes are `XboxControl` and `PoseControl`.

XboxControl subscribes to a joy topic and controls the velocity of each joint directly based on the values for each controller axis. Each controller axis is represented as a value scaled from -1 to 1. Each controller axis value is multiplied by the maximum motor velocity, scaling it linearly. Motor velocities are assembled and published via the RobolinkControl message. In order to increase user safety, a "dead man" switch was implemented that requires the user to hold a button down while controlling the arm.

The RobolinkControl message is a custom ROS message used to dictate the mode in which the robolink is controlled. The message consists of eight fields. The first field, control_mode defines which desired method of control is tu be used. control_mode can be one of five different modes: joint velocity control allows direct control of each joint's velocity in RPM, joint absolute position control allows direct control of each joint's absolute position in encoder ticks, joint relative position allow direct control of each joint's relative position, pose control allows control of the gripper's cartesian position, and twist control allows control of the gripper's cartesian velocity. The next five fields are used to define joint setpoints for direct control. The seventh and eighth fields are standard pose and twist messages respectively.

PoseControl is used to execute cartesian position and velocity control of the end effector. PoseControl publishes a pose in the arm's gripper frame to the RobolinkControl topic. Publishing a pose in the gripper frame increases the modularity of the code by allowing higher level controllers to dictate the pose of the gripper without needing to calculate the transfomations between the gripper and the controller's base frame.

The RobolinkDriver node is responsible for all forward and inverse kinematics of the arm. The node subscribes to the RobolinkControl topic and controls the arm in the mode defined in the message. Directly controlling the arm joints is accomplished by building and publishing a GroupEPOSControl message with the specified setpoints. Cartesian control of the gripper is accomplished via a PID controller. Figure 6 shows the flow of the pose controller. The desired pose is transformed to the frame of the arm base and cartesian tip error is calculated. Tip error is then passed to the PID controller and the resulting tip velocities are translated into joint velocities via the pseudo inverse of the jacobian. Because of the limited joint angles and degrees of freedom of the robolink, orientation is ignored in the pose controller. This is accomplished by only using the top half of the jacobian during translation.

In order to keep the jacobian as up to date as possible. The Robolink Driver node subscribes to the Group_Motor_Info topic. Every time a message is recieved on this topic, joint angles are updated, transformations are recalculated, and the jacobian is updated. The RobolinkDriver node also publishes a RobolinkInfo message containing the current pose of the arm, current joint positions, and current joint velocities.

At the lowest level, EPOSManager initializes and

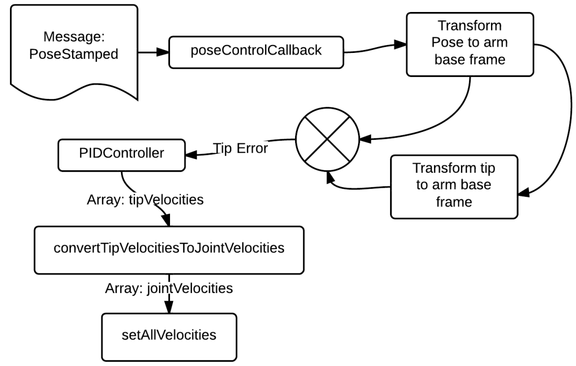

Fig. 6. Flow of pose control for the robolink.

communicates with the individual EPOS controllers. The GroupEPOSControl message is used to control the motors. The GroupEPOSControl message is a list of EPOSControl messages, each of which contain the CAN node of the motor to be controlled, the control mode for the motor, and the motor's setpoint. EPOSManager also collects the reported information from each controller (motor position, motor velocity, motor current, and any status codes), creates a list of MotorInfo messages and publishes it to the GroupMotorInfo topic.

The Linux library for the EPOS controllers does not support a watchdog timer. To remedy this, a higher level watchdog timer was implemented that stops all joint motion if a RobolinkControl message is not received for three seconds. A low level monitor was implemented to prevent damage to the robolink in the event of a command causing the joint to travel beyond its limits. If a joint is at its minimum or maximum value, the monitor will set joint velocities in that direction to zero regardless of the control signal received.

III. Results

The final system can be seen in Figure 7. The drive module is fits within a 12in x 12in x 4in box and houses all motors, controllers, and connectors for the robolink. A final end-effector was not chosen for the robolink as future work will be done regarding passive end-effectors.

Figures 8 and 9 show plots of joint angles and velocities as pose control is used to move the end effector in a .2 meter long straight line. Open loop control was implemented using forward kinematics to estimate gripper position and the inverse jacobian to generate tip velocities toward a goal position. The arm's initial configuration was folded and the straight line it followed caused the arm to straighten out. This is reflected in Figure 8 as joints 1 and 3 can be seen trending toward zero and the arm's zero state is fully straight. As the arm straightened, joint velocities needed to increase because the vector produced by joint movement was getting smaller. This is reflected in Figure 9. Small inconsistencies can be seen in the joint velocities as the continuously updated jacobian adjusts to follow the trajectory.

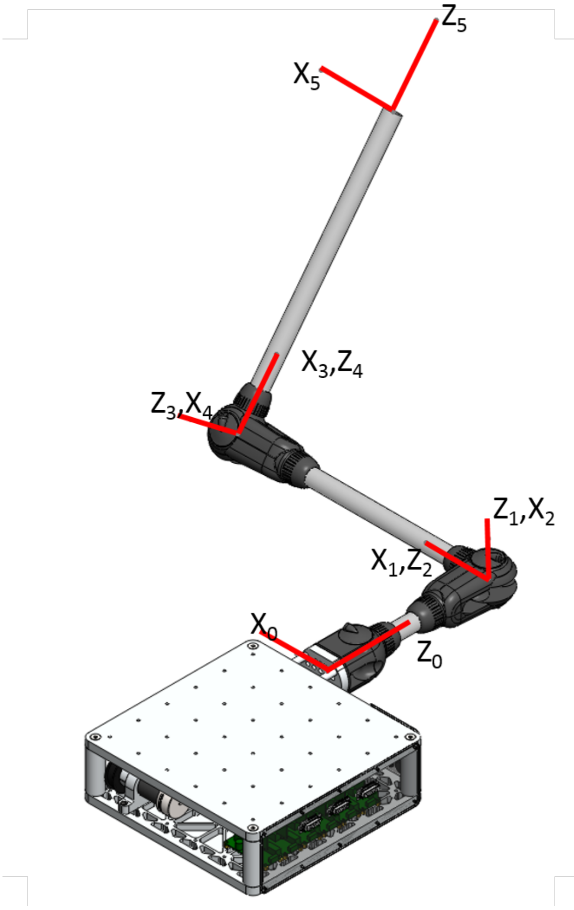

Fig. 7. The WPI Robolink system.

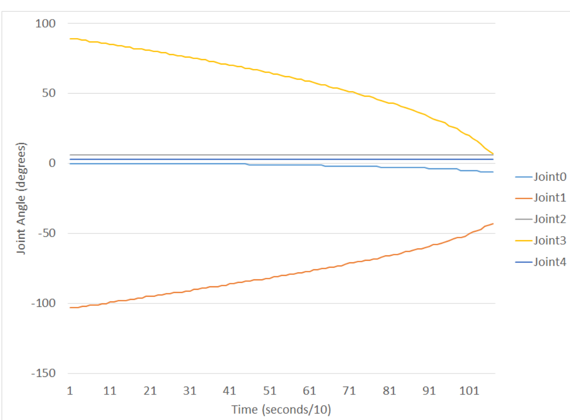

Fig. 8. Joint Angles as tip follows a .2 meter long straight line.

IV. Conclusion

There is a need for inexpensive semi-autonomous systems to assist those who cannot perform activities of daily living. An modular robotic arm was developed for use with a human-in-the-loop cyber physical system. Choosing a commercially available cable driven arm reduced costs and allowed for the development of a modular drive module. The module

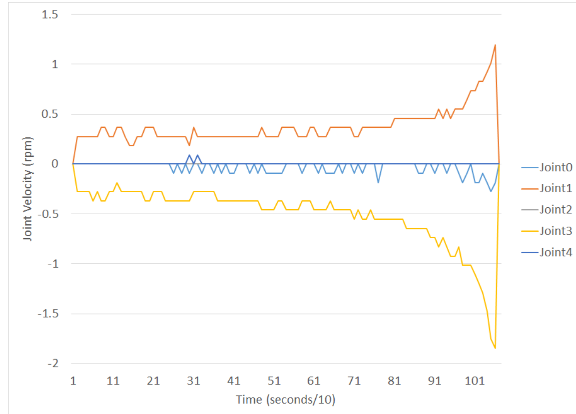

Fig. 9. Joint Velocities as tip follows a .2 meter long straight line.

is controlled by an external computer and utilizes the ROS framework. Control of the arm can be accomplished by both direct control of the joints and by cartesian pose control of the end effector. Cartesian control is capable of following a trajectory using open loop control.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Award No. 1135854. The authors would also like to acknowledge the support of igus, and Maxon Motors.

References

[1] A. D. Wood and J. A. Stankovic, “Human in the loop: distributed data streams for immersive cyber-physical systems,” *SIGBED Rev.*, vol. 5, no. 1, pp. 20:1–20:2, Jan. 2008. [Online]. Available: http://doi.acm.org/10.1145/1366283.1366303

[2] “NIST Foundations for innovation in cyber-physical systems workshop summary report,” 2012, http://events.energetics.com/NIST-CPSWorkshop/pdfs/CPS_WorkshopReprot_v6_6-12-12_DRAFT.pdf.

[3] G. Schirner, D. Erdogmus, K. Chowdhury, and T. Padir, “The future of human-in-the-loop cyber-physical systems,” *IEEE Computer*, vol. 46, no. 1, 2013.

[4] F. Galan, M. Nuttin, E. Lew, P. Ferrez, G. Vanacker, J. Philips, and J. del R. Millan, “A brain-actuated wheelchair: Asynchronous and non-invasive brain-computer interfaces for continuous control of robots,” *Clinical Neurophysiology*, vol. 119, no. 9, pp. 2159 – 2169, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1388245708005750

[5] J. d. R. Millan, R. Rupp, G. Muller-Putz, R. Murray-Smith, C. Giugliemma, M. Tangermann, A. Kubler, R. Leeb, C. Neuper, K. R. Muller, and D. Mattia, “Combining Brain-Computer Interfaces and Assistive Technologies: State-of-the-Art and Challenges,” *Frontiers in Neuroprosthetics*, 2010.

[6] R. C. Simpson, “Smart wheelchairs: A literature review,” *Journal of Rehabilitation, Research & Development*, vol. 42, no. 4, pp. 423–438, Jul. 2005.

[7] H. Krebs, “Rehabilitation robotics: Performance-based progressive robot-assisted therapy,” *Autonomous Robots*, vol. 15, pp. 7–20, 2003.

[8] K. M. Tsui, H. A. Yanco, D. J. Feil-Seifer, and M. J. Matarić, “Survey of domain-specific performance measures in assistive robotic technology,” in *Proceedings of the 8th Workshop on Performance Metrics for Intelligent Systems*, ser. PerMIS ’08. New York, NY, USA: ACM, 2008, pp. 116–123.

[9] H. Yanco, "Wheelesley: A robotic wheelchair system: Indoor navigation and user interface," in *Assistive Technology and Artificial Intelligence*, ser. Lecture Notes in Computer Science, V. Mittal, H. Yanco, J. Aronis, and R. Simpson, Eds. Springer Berlin / Heidelberg, 1998, vol. 1458, pp. 256–268, 10.1007/BFb0055983. [Online]. Available: http://dx.doi.org/10.1007/BFb0055983

[10] F. Cheein, R. Carelli, C. De la Cruz, S. Muller, and T. Filho, "Autonomous assistance navigation for robotic wheelchairs in confined spaces," in *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, Sep. 2010, pp. 503 –506.

[11] F. Auat Cheein, C. De La Cruz, R. Carelli, and T. Bastos-Filho, "Solution to a door crossing problem for an autonomous wheelchair," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, Oct. 2009, pp. 4931 –4936.

[12] A. Lopes, U. Nunes, and L. Vaz, "Assisted navigation based on shared-control, using discrete and sparse human-machine interfaces," in *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, Sep. 2010, pp. 471 –474.

[13] S. P. Parikh, V. Grassi Jr., V. Kumar, and J. Okamoto Jr., "Integrating human inputs with autonomous behaviors on an intelligent wheelchair platform," *Intelligent Systems, IEEE*, vol. 22, no. 2, pp. 33 –41, Mar. 2007.

[14] I. Iturrate, J. Antelis, A. Kubler, and J. Minguez, "A noninvasive brain-actuated wheelchair based on a p300 neurophysiological protocol and automated navigation," *Robotics, IEEE Transactions on*, vol. 25, no. 3, pp. 614 –627, Jun. 2009.

[15] X. Perrin, R. Chavarriaga, F. Colas, R. Siegwart, and J. del R. Millan, "Brain-coupled interaction for semi-autonomous navigation of an assistive robot," *Robotics and Autonomous Systems*, vol. 58, no. 12, pp. 1246 – 1255, 2010, intelligent Robotics and Neuroscience.

[16] R. Desmond, M. Dickerman, J. Fleming, D. Sinyukov, J. Schaufeld, and T. Padir, "Development of modular sensors for semi- autonomous wheelchairs," in *Proc. 2013 IEEE International Conference on Technologies for Practical Robot Applications (TePRA)*, April.