A Wireless Test-bed for Modelling and Validation of Electric Powered Wheelchairs

Xiaoqi Chen(1), Patrick Wolm(1), Isaac Anstis(1), John Oldridge(1), William Hanbury-Webber(1), Rodney Elliot(1), J Geoff Chase(1), Warren Pettigrew(2)

(1)Dept of Mech Engrg, Univ. of Canterbury, Private bag 4800Christchurch 8140, New Zealand
xiaoqi.chen@canterbury.ac.nz

(2)Dynamic Controls Ltd, 17 Print Place, Middleton, Christchurch 8015, New Zealand
wpettigrew@dynamiccontrols.com

Abstract

A wireless test-bed has been developed to model electric powered wheelchairs. The test-bed integrates sensors, embedded controller, and the motorised mechanical system. Real time data acquisition and analysis is performed in dSpace. Simulink is used to build the computer model of the wheelchair. The wireless communication, based on Bluetooth technology, ensures the integrity of sensor data collected and control signals sent. The developed test-bed not only facilitates parameterisation, modelling and model validation of motorised wheelchairs, and wheeled mobile robots in general; but also serves as an excellent mechatronics teaching platform.

Keywords: electric powered wheelchair, test-bed, virtual model, wireless communication

1 Introduction

Future electric powered wheelchair should meet the needs of a large user base of aged and disabled people. Intensive research efforts have been put in to develop highly effective home-installed devices which will provide the user with continuous assistance in achieving high mobility without compromising comfort. Linden et al [1] surveyed the transportation needs of individuals with disabilities. Modern designs demand that the user is less engaged in the control process, and the interaction between the device and human-friendly. In other word, the devices should possess a high level of intelligence in their controls, actions, and interactions with the user, offering him/her a high level of comfort and functionality. Stefanov et al [2] considered a mixed control mode as a better alternative for human-machine interactions for users with great movement disability.

Kamper et al [3] examined the posture of wheelchair users experiencing external perturbations through the use of a tilt platform. Their work revealed that spinal cord injury subjects lost balance at perturbation levels seen during normal braking maneuvers.

Boiadzjiev & Stefanov's work [4] studied powered four-wheeled wheelchairs with two driving wheels and two small sized front wheels. The proposed controller includes a special module that analyses user's commands before their execution.

Boquete et al [5] studied the problem of controlling the movements of a handicapped person's motorized wheelchair from a practical point of view. Low level PID controller directly controls the drivers of the chair's motors, with a classic PID (proportional-integral-derivative) control loop. The high level control was implemented by means of neural techniques to ensure linear and angular speeds of the wheelchair follow those indicated by a trajectory generator. Ding et al [6] examined the influences of caster orientations and driving speeds on the reverse control of electric-powered wheelchairs.

In order to prototype wheelchair control functions, man-machine interface, and predict the performance of wheelchair, a virtual model of the wheelchair dynamics need to be created. This model can be further validated quickly and repeatedly, without the difficulties of physically setting the wheelchair up for each test. This requires a robust wireless test-bed which transmits sensor data and receives motor control instructions without data corruption and timing issues.

This work presents a wireless test-bed which allows modelling and model validation of electric powered wheelchairs. It firstly discusses the key system components: the hardware and software platform, data flow, sensors integrated in the test-bed, and sensor data transmission with an in-house build microcomputer. Wheelchair models are developed with Simulink. Bluetooth has been adopted as the wireless data link. Three types of Bluetooth modules have been used in the wireless test-bed, namely, sensor data communication via a pair of Brain-boxes BL-819 models, control data communication using low-cost USB Class One adaptor, and dSpace data communication through serial channels installed in the dSpace computer. The integrated system provides the flexibility for the dynamics of electric wheelchair to be modelled and validated. It also serves an excellent mechatronics teaching platform where sensors & instrumentation, mechanical characterisation, modelling, control development can be experimented.

2 Mechatronics Approach

2.1 System Components

Figure 1 shows the top level view of the wireless test-bed of the wheelchair control prototyping. Key components of the integrated mechatronics system include hardware and software platform, wireless communication, sensors and motor control.

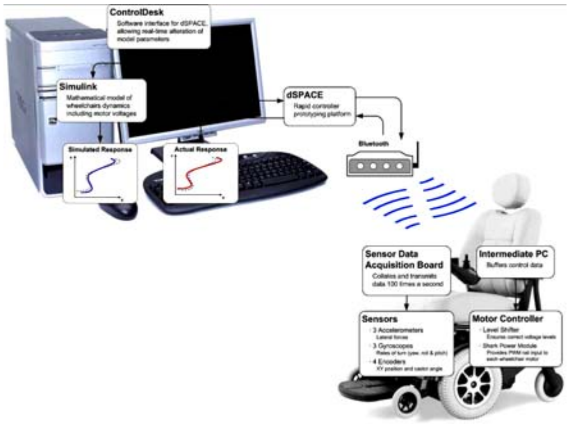

Figure 1: Wheelchair System Overview

dSPACE and **ControlDesk**: The hardware and software package is the cornerstone of the whole wireless platform. It is through this system that real time data acquisition and analysis is performed. ControlDesk also enables the parameters, such as the input motor voltages, to be varied in real time.

Simulink: Simulink, a simulation program within Matlab, was used to build the computer model of the wheelchair. It was interfaced closely with ControlDesk for testing.

Bluetooth: Bluetooth was employed for wireless communications between the wheelchair and base PC. The use of a wireless data link enabled the wheelchair to move freely without constriction and also avoided noise and interference problems which had been encountered a wired link.

Sensors: The sensors onboard the wheelchair includes gyroscopes, accelerometers and rotational encoders. They provide feedback information regarding the wheelchairs speed, position and forces exerted upon the wheelchair.

Shark Power Module (SPM): SPM was the motor controller provided by Dynamic Controls. It relays the commands, with appropriate control and protection, to the DC motors of the wheelchair. This compact module is located on the wheelchair. The **SharkBus** controls the information flow on a strict time cycle for the SPM, i.e. all data flow to and from a remote controller such as a joystick, all the data flow within the SPM and from the SPM to the motors.

2.2 Dataflow Path

Data flow is serial in nature and is based on the RS232 standard. Initially, an attempt was made to communicate directly between dSPACE and the SPM; but involving extensive software development. Instead an API, know as a Dynamically Linked Library or DLL (*SharkAPI.dll*) was used to implement the communication protocol, as shown in **Figure 2**. It proved to be invaluable once implemented correctly.

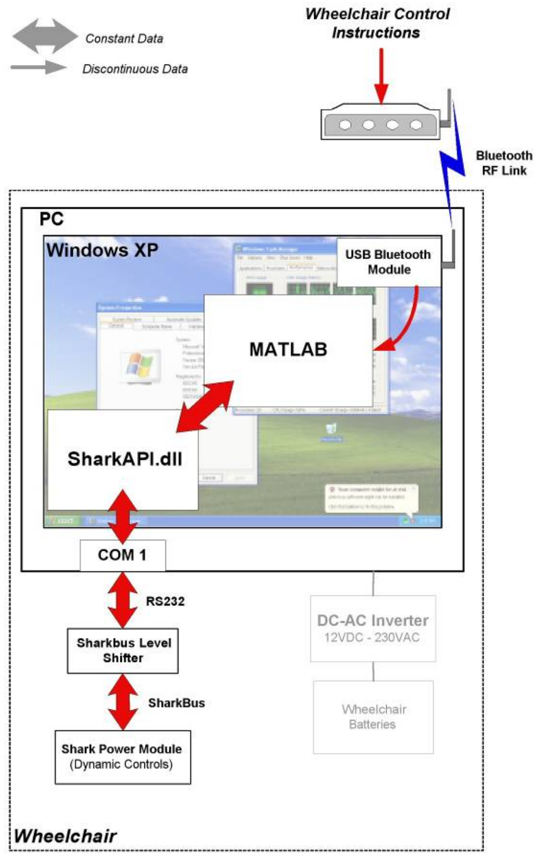

Figure 2: Wheelchair Control Dataflow Path

Once the API was running (directly through a serial cable), attempts were made to direct the data from the API out through Bluetooth to the level shifter utilising Matlab M-files. Unfortunately, the SharkBus protocol is very demanding on specific *20 ms* timings, and this proved to be impossible to run, as the Bluetooth modules would put an indeterminate delay of between *5* to *20 ms* into each transmission, which falls well outside of the SharkBus allowances.

This demanded that a flexible interface that would account for the Bluetooth's latency issue. The solution was to install a Windows-Based Mobile PC on the wheelchair. The Mobile PC with API acts as a pseudo hold buffer, waiting till all the available data has arrived before passing it onto the SPM. There are alternatives to this, such as using a smaller PDA or run a custom microcontroller (which would require the entire SharkBus protocol to be rewritten for the Microcontroller to run).

C++ Win32 API has been integrated to handle all low level communication between the wheelchair and the PC. To implement this, a simple program, running inside the Matlab environment, acts as a relay between DSPACE and the SharkBus. The main tasks of the program include:

- Open Bluetooth serial connection.
- Start the wheelchair up.
- Listen to the Bluetooth serial port and wait for data.
- Interpret the control data sent from the DSPACE system.
- Issue commands to the wheelchair power module.

2.3 Sensors

Sensors are required to provided motion and force information fro closed-loop control of the wheelchair, **Figure 3** shows various sensors mounted in the wheelchair, including three accelerometers; three gyros; four rotational encoders. Two of the accelerometers are placed adjacent to the front wheels to measure the side and driving force on the wheels. These front two accelerometers were mounted on aluminium brackets, designed to get the accelerometer chip itself in line with the drive wheel axes and as close to the centre of the wheel as possible. The third accelerometer was placed on the back support between the castor wheels to measure forces on the casters and help determine wheelchair skid or over-steer.

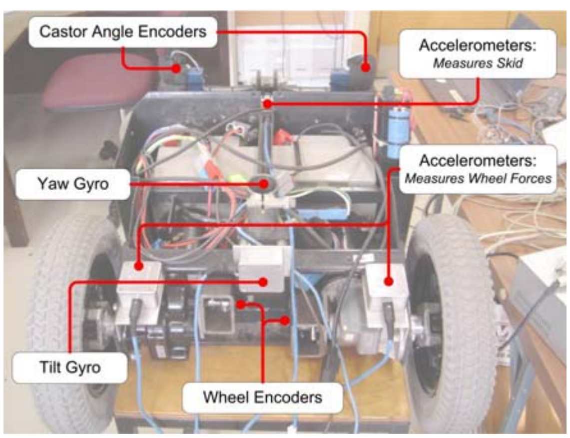

Figure 3: Sensor Placement

Two of the gyros were placed to measure the tilt (sideways tip of the wheelchair) and yaw (turning rotation of the wheelchair about its vertical axis). For the gyro intended to measure tilt, a bracket was made, sitting in the centre between the drive wheels. It holds the gyro so that its rotational axis is in line with the direction of travel. The gyro to measure yaw can be set anywhere along the vertical axis passing through the Centre of Gravity (COG). The third gyro preferably is mounted exactly at the COG and is for future expansion and could be used to help determine jerk, or the degree to which the wheelchair rocks backwards and forwards.

The front two wheel encoders are for experimental purposes to give a relative XY position and path over the ground. The rear encoders on the castors are necessary when the model starts to include effect of the castor angle. Encoders were also installed for the front wheels. All encoders were mounted in custom built housings.

In addition to the wheelchair components itself, the wheelchair needs to also carry several items to provide wireless capabilities. These include sensor data acquisition box, the Shark power module, the Bluetooth modules box and the lever-shifter housing. Housing was designed to house these items permanently, conceal the batteries and some of the wirings.

2.4 Sensor Data Transmission

Data from the onboard sensors is collated and sent wirelessly using a microcontroller unit.

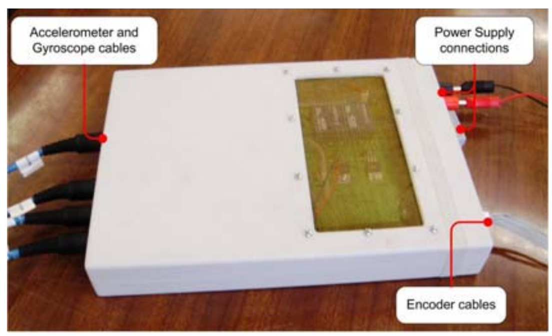

Figure 4: Sensor data processing/transmitting box

The in-house design-and-built unit, shown n **Figure 4**, receives data directly from the sensors and transmits via Bluetooth at approximately *100 Hz*. The sensor data is transmitted in a 46 byte package per second, with the first 44 bytes containing the collated gyro, accelerometer and encoder data and the remaining 2 bytes a new-line and a carriage return character.

The sensor data was collated in a set order with the first 12 bytes containing the gyro values (4 bytes per gyro), the next 24 bytes the accelerometer values (8 bytes per accelerometer, 4 for the *x* direction, 4 for the *y* direction), and the last 8 bytes represented the rotational encoders (2 bytes each). The order of the gyros and accelerometers within their relevant section of the 44 byte data package are given by the numbers on the cables and the plug slots on the back of the sensor data processing/transmitting box.

The transmitted sensor data was captured in real time through ControlDesk, and was saved automatically to a Matlab comma-separated numbers (.csv) file. Matlab M-files were written to decode and interpret the data. The data processing involves four basic steps:

1) Retrieve the useful data from the .csv file which is created in real time through ControlDesk.
2) Parse and separate the byte packages which represent the individual gyro, accelerometer and encoder data from the main 44 byte package.
3) Change the individual sensor data packages to a binary string.
4) Convert the binary to 32-bit single precision floating integers using the IEEE standard for floating point numbers, for the gyros and accelerometers, and 16-bit unsigned integers for the encoders.

3 Wheelchair Model

The mathematical model of the wheelchair takes as inputs the voltages to the two driving motors and outputs the position of the chair in the X-Y plane. The advantages of having an accurate software model of the wheelchair can not be understated, and once validated will be an invaluable tool for rapid prototyping of a control system.

The choice to use Simulink as the platform to develop the model was due to two reasons. Firstly, Simulink is the dedicated platform of choice for interface with dSPACE through ControlDesk. Secondly, and most importantly, the equations derived could not be put in State Space form, as initially desired, due to non-linear sinusoid coefficients. State Space form would have enabled the use of simpler Matlab M-files and more advanced feedback control options.

Following research into vehicle dynamics, two separate basic models were created. The first model, a purely kinematic model, simply treats the wheelchair as two wheels separated at a distance. It is governed by Equations (1) to (3) as follows.

$$\dot{\theta} = \frac{r}{d_2}(\omega_1 - \omega_2) \tag{1}$$

$$v_u = \frac{(\omega_1 + \omega_2)r}{2} \tag{2}$$

$$v_w = \dot{\theta} d \tag{3}$$

$\dot{\theta}$ is the yaw rate, v_u and v_w the forward and sideways speeds of the wheelchair respectively, and $\omega_{1,2}$ the rotating speeds of the front wheels.

The above kinematic model proved to be robust and seemingly accurate. But it was not expandable to meet the requirements of the control prototyping application.

The second prototype model treats the wheelchair as a two dimensional object, and focuses on the forces and moments acting on the chair. It proved difficult to get the model to follow an expected path. However its ability to be adjusted made it the preferred choice for developing the control prototyping system further. Another advantage of the force model was that the data available from the accelerometers and gyros, in particular the forces on the wheels, could be compared directly to parts of the model.

The model was initially based around a simple free body diagram involving two dimensional forces on the driving wheels and the centre of gravity, which was assumed to be in the middle of all four wheels. Over time, additions were made such that the final free body diagram evolved into that pictured below. The model was further enhanced to include forces on both driving wheels and passive wheels, and to allow for the offset of COG from the geometric centre of four wheels, as illustrated in **Figure 5**.

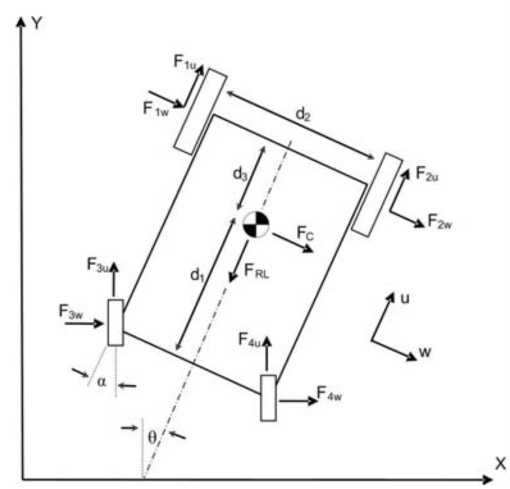

Figure 5: Free Body Diagram of Wheelchair

By summing all forces in both the local *u* and *w* directions, as well as taking moments about the centre of gravity, both rotational and translational accelerations are determined.

The driving forces on the two front wheels, F_{1u} and F_{2u}, were calculated using the DC motor equations. These equations relate the voltage across the motors to the output torque on the wheels. Knowing the torque, τ , the linear force, F , can easily be deduced using Equation (4)

$$F = \frac{\tau}{r} = \frac{I\alpha}{r} = \frac{I\ddot{\theta}}{r} \tag{4}$$

Modelling the side forces on the wheels was not as straight forward as expected. When travelling in a straight line, these forces equal zero and when cornering they are related to how sharp and fast the turn is. For modelling purposes, the centripetal force acting through the centre of gravity was split into four forces acting on each wheel depending on their distance to the centre. This centripetal force is directly proportional to the yaw rate, $\dot{\theta}$, and the forward speed of the chair, v_u .

$$F_c = \frac{mv_u^2}{r} = mv_u\dot{\theta} \tag{5}$$

where *m* is mass.

In doing so, the cornering of the model wheelchair proved very lifelike.

The actual voltages going to the wheelchair motors are pulse width modulated (PWM) signals. The signal possesses a constant frequency of *20 kHz* and a *12 V* amplitude, with the motor speeds varied by changing the duty cycle. A novel approach was developed to create a PWM signal using standard Simulink library blocks. By offsetting a triangular wave up and down the Y–axis and then taking the sign of the result, a simple PWM signal was created whose duty cycle could be controlled. **Figure 6** details the implementation of the PWM generator in Simulink.

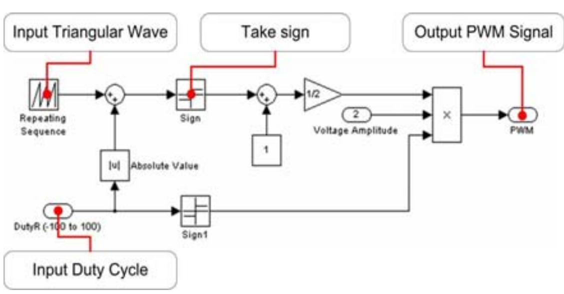

Figure 6: Simulink PWM generator subsystem

4 Wireless Data Link

Bluetooth is one of the newest short range wireless links, which has been designed for low bandwidth, short range transmission, based on low cost transceivers. It is becoming increasingly adopted by many wireless PC-based systems. Bluetooth suits this application which does not require a large amount of data to be transferred.

However, the latency of the Bluetooth link, the specification of which was difficult to find, became a concern. After some experimentation and a lot of research, it was shown that there is a transmission delay of about *5 – 20 ms*. This eliminates the ability to use Bluetooth directly between the computer and the Level Shifter. Clearly a buffer of some description was needed, which called for a Mobile computer mounted on the wheelchair to serve as communication buffer. In this research, three types of Bluetooth modules were used for sensor communication, control data communication and dSpace data communication respectively.

4.1 Sensor Data Communication

The data from each of the sensors is passed through to the DSPACE system, via a pair of Brain-boxes BL-819 modules shown in **Figure 7**.

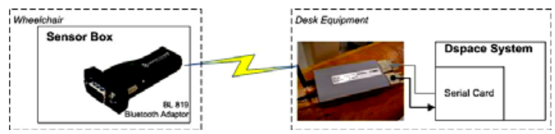

Figure 7: Sensor Data Communication Path

These units are a self contained Class 2 Bluetooth RS232 Module, which have onboard flash memory to store all serial settings. One of the key features of these modules, which are not found on many other units, is the point-to-point connectivity. Once set up, the modules are completely transparent, and appear as a serial cable, and are not discoverable to any other Bluetooth modules. One of these modules is located inside the sensor box on the wheelchair, the other in the Bluetooth Serial Communication box.

4.2 Control Data Communication

A Class 2 Bluetooth RS232 adaptor has a range of approximately 10 metres. It suffices for sensor data communication. For the control data, approaching the limit of the range, such a adaptor risks data packets being dropped, or possibly loosing connectivity completely. To avoid this problem, a Class One adaptor is used, which has a range of approximately *100 m*. A low-cost USB Class one was connected to the mobile PC which handles high level communication, shown in **Figure 8**.

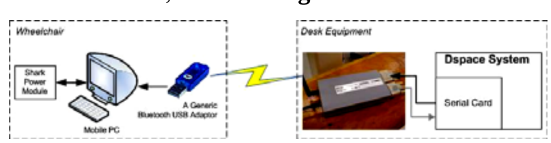

Figure 8: Control Data Communication Path

To act as a Bluetooth Serial Port, a module from SparkFun called the "BlueSmirf" was used. This is a Class One wireless serial cable replacement that works as a serial (RX/TX) pipe. Any serial stream from 9600 to 115200bps can be passed seamlessly from to the computer to dSPACE and vice versa. This has been set up for the same parameters as in the sensor date communication the module.

However the control data communication module does not provide RS232 voltages. It only provides TTL levels, which are boosted to the required levels with a simple MAX232 Circuit.

4.3 dSPACE Data Communication

Transmission and receipt of data for real time applications are achieved through the serial card (*ds4201S*) installed on the dSPACE computer. The data transmitted is the duty cycle settings for both the left and right DC motors of the wheelchair. **Figure 9** shows the Simulink transmission model that is compiled into C language, then loaded into the dSPACE computer where it operates independently and is controlled through the ControlDesk graphical use interface (GUI). The build process is then shown in the Matlab workspace.

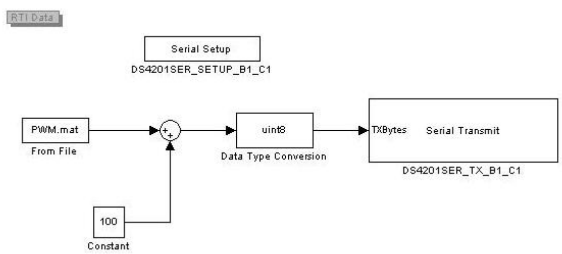

Figure 9: Simulink model to transmit motor control values

The *ds4201S* serial card only transmits and receives unsigned 8 bit integers with a range of *0* to *255*, hence the Data Type Conversion. Since duty cycles can be negative numbers, *100* is added to their values to ensure full transmission of the data otherwise negative numbers are treated as *0*.

Figure 10 shows the receiving set up which is very similar to the transmit model.

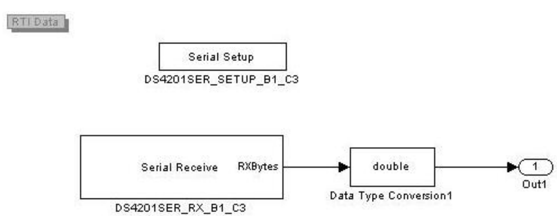

Figure 10: Simulink model used to receive sensor data

The same procedures are followed, as in the transmit model, to load the receive model into dSPACE.

5 Conclusions

A wireless test-bed, based on Bluetooth technology, has been developed to model electric powered wheelchairs. It overcomes the difficulties otherwise experienced with wired connections. Solid state sensors such as accelerometers, gyroscopes and rotational encoders are embedded into the test-bed to collect position and motion date in real time. Real-time data acquisition, data processing and interpretation are carried out in the dSpace and ControlDesk.

By instituting a wireless test-bed, the corruption of control signals sent or sensor data received is eliminated. Restrictions on motion and the running of the wheelchair to its limits to obtain effective data are also removed with the elimination of the cable tether. With the creation of the dynamic model and wireless test-bed the model can be validated rapidly. The integrated mechatronics system also provides an excellent teaching platform to conduct modelling of electromechanical system, instrumentation, and actuator control.

6 Reference

[1] M. Linden, D. Kamper, S. Reger, T. Adams, “Transportation needs: survey of individuals with disabilities”, *Proceedings of the 19th Annual RESNA Conference;* 1996 June 7-12, Salt Lake City, UT. Washington, DC: RESNA Press, pp. 52-54 (1966).

[2] D.H. Stefanov, Z. Bien, W.K., “Some aspects of human-friendly control for movement-helping devices”, *Journal of Artificial Life and Robotics*. Issue Volume 4, Number 4 / December 2000, pp. 198-205 (2000).

[3] D. Kamper; M. Parnianpour; K. Barin; T. Adams; M. Linden; H. Hemami, “Postural stability of wheelchair users exposed to sustained, external perturbations”, *Journal of Rehabilitation Research & Development*, Vol. 36 No. 2 (1999).

[4] G. Boiadzjiev, D. Stefanov, “Powered weelchair control based on the dynamical criteria of stability”, *Mechatronics*, Volume 12, Number 4, May 2002, pp. 543-562(20) (2002).

[5] L. Boquete1, R. García1, R. Barea1 and M. Mazo, “Neural Control of the Movements of a Wheelchair”, *Journal of Intelligent and Robotic Systems*, Issue Volume 25, Number 3, pp. 213-226 (1999).

[6] D. Ding, R. A. Cooper, S. Guo, T.A. Corfman, “Analysis of Driving Backward in an Electric-Powered Wheelchair”, *IEEE Transactions on Control System Technology*, Vol. 12, No. 6 (2004).