Abstract—This paper deals with various ways of controlling an electrically powered wheelchair beyond the usual method involving a manual joystick. The main focus is on the newest version of HaWCoS – the "HAnds-free Wheelchair COntrol System" – allowing persons with severe disabilities to reliably navigate a power wheelchair without the need to use the hands. All the user has to do is to produce a sequence of tiny contractions of an arbitrary muscle, e.g., by raising the eyebrow. The working prototype of the system, which has been realized as a stand-alone device, is introduced in detail, together with a closer look at the muscle-based control principle and a brief description of a PC-based simulator. Finally, the advantages and the drawbacks of the system are discussed on the basis of a rather simple real-life experiment.

I. INTRODUCTION

Using an electrically powered wheelchair can mean a tremendous amount of independence for someone with a severe physical disability who can neither walk nor operate a "mechanical" wheelchair alone. The problem is that in many cases the disability causing that someone not to be able to walk also keeps her/him from reliably employing the hands.

Unfortunately, the standard way of controlling a power wheelchair is by operating a joystick, and it therefore does require certain "manual skills". As a consequence, thinking about hands-free alternatives in this context of course appears interesting and promising, but most of all it is necessary.

A common way to realize such an alternative control method is to monitor a specific bio-signal related to a certain bodily function of the user – such as brain-waves, muscular activity, or eye posture – and to react to the detection of a particular pattern in the monitored signal. By willfully generating one of those patterns, the user can thus trigger the corresponding reaction, i.e., issue control commands (see [1]).

The following section presents a number of hands-free systems – either in use today or (based on various bio-signal interfaces) reported in the literature. The main part of this paper is concerned with the newest version of the "HAnds-free Wheelchair COntrol System" (HaWCoS), which is based on the detection of intentional muscle contractions.

Unlike its predecessor (introduced in [2]), the new (extended and enhanced) implementation of HaWCoS has been realized as a stand-alone device which does not depend on the use of a laptop computer. After a basic reiteration of the muscle-based control principle the hardware of the stand-alone device as well as the underlying software are presented in thorough detail. To make a copy of the prototypical device readily available to, for instance, the wheelchair manufacturer, the system has also been implemented as a Windows® executable with identical program behavior (running on any standard laptop) – a short description of this simulator follows the introduction of the stand-alone device.

In order to demonstrate the feasibility of the new realization of HaWCoS, a simple real-life experiment has been conducted. In addition to the mere numerical results, benefits of the system are listed, and problems are relentlessly discussed.

Finally, the paper is concluded with a brief summary and a quick look into the future.

II. RELATED WORK

When engrossing the hands is not permitted, the most straightforward ideas relate to the question: "What other part of the body can be substituted to operate some sort of a proportional joystick?" Answers to this question lead to chin-control modules (see fig. 1a), "mouthstick" controllers (see fig. 1b), sip-and-puff products, systems based on head movements (see [3]), or tongue-operated solutions (see [4]) – each of those can in some form be found on the market nowadays.

![Fig. 1. Currently Available Specialty Controls: a: Chin-control; b: "Mouthstick" (reprinted courtesy of MEYRA)](image-url)

As far as scientific research is concerned, dealing with bio-signal systems (which, as already mentioned in the introduction, concentrate on the time series of a suitable bodily function of the user) has become more and more popular recently. When talking about assisting people with physical disabilities, the bodily function that is affected by mental activity only, i.e., brain-waves, seems to be ideal.

Systems which particularly focus on this type of time series (e.g., monitored in the form of an EEG recording) are called brain-computer interfaces or BCI's (e.g., [5], [6]). A BCI examines the on-going EEG of a subject and looks for...
recurring patterns, mostly with the help of a neural classifier (e.g., [7]). To do this, the time series is chopped up into smaller portions (or samples) which are then fed into the network.

Unfortunately, applying the BCI idea to a wheelchair control system is virtually impossible. The main reason is the sensitivity of the EEG signal. Whenever the subject blinks, swallows, laughs, talks, or moves in any way, the corresponding EEG sample is rendered unusable, since the EEG consists of very tiny voltage potentials, that have to be amplified by a factor on the order of $10^4$ – but any noise contamination gets amplified, too (see also [8], [9]). Furthermore, since a wheelchair driver constantly moves with the wheelchair he or she is sitting in – e.g., trying to compensate braking and/or turning – there would inevitably be a lot of contamination!

Besides, EEG-based BCIs are awfully slow: [10] talks about an achievable input rate of up to 25 bits per minute, which is less than 1 bit every 2 seconds. In other words: it would take more than 2 seconds to stop the wheelchair in case of a sudden obstacle (provided a brain-controlled "yes/no" switch were used) – this is totally unacceptable as the basis of a meaningful real-time control system. Therefore, BCI’s may well be used for controlling the cursor on a computer screen [11], for moving a graphical object through a virtual environment [12], or for remotely operating a mobile robot (that’s what is actually reported in [13]), but not for really driving in a wheelchair (see also [14]; in the experiment conducted there, the chair is stopped during EEG analysis).

Other bio-signal interfaces process EO-s signals which document eye movements (e.g., [15]) or EMG signals which record muscular activity (e.g., [16]). EO-s-based systems have the disadvantage that they rely on the eyes as input medium, so the user is not totally free in looking wherever he or she wants while driving – this is extremely questionable and might potentially result in hazardous situations. Muscular activity is arguably the most suitable input source for wheelchair control. However, common EMG-based systems are also relatively error-prone (not as bad as BCT’s, though) and they often ask for considerably complicated muscular interaction.

In addition to bio-signal interfaces, a lot of hands-free wheelchair control approaches use voice commands as input signals (e.g., [17]), mostly for retrieving the desired basic movement direction. The disadvantage of those systems is that they are unsuitable for loud and noisy environments, e.g., during parties. Other, more sophisticated (but also more expensive) approaches involve video cameras and computer vision, as well as combinations of different types of input signals (e.g., [18], [19]).

The HaWCoS system, which is the solution to the control problem detailed in this paper, is based on muscle-related input. However, it does not rely on common EMG electrodes, but instead uses a piezo-based sensor (the advantage of this setup is explained in the next section). As becomes clear below, when comparing HaWCoS’ user interface to an EEG-based BCI, it has to be admitted that it asks for slightly more physical input, while its sensitivity to noise is not even anywhere near.

### III. Muscle-related Input Method

In a perfect world, it would be ideal to monitor the brainwaves of someone with a severe disability and to use this as the input to an interface allowing that person to drive around in a power wheelchair. That way, everyone with a clear mind could use this interface, even a completely locked-in patient, who has retained no physical ability whatsoever. Unfortunately, the world is not that perfect, and since EEG recording is awfully sensitive to noise, using it for an application like this is simply impossible.

Therefore, a compromise has to be made. The goal must be to devise an interface which requires as little physical contribution as possible from the user, while still being robust and reliable enough to control the wheelchair the user of the interface is sitting in. One possible solution is represented by the input method adopted in HaWCoS, which is the topic of this section.

The basic idea is to look for intentional contractions in the activity of a muscle of the user and to generate control commands according to the temporal sequence of those contractions. The reason why counting on muscular activity seems to be favorable is that many persons with disabilities – even when otherwise completely paralyzed – have retained reliable control over at least one (sometimes the strangest!) muscle group, e.g., a facial muscle, a shoulder muscle or even a single finger on the left hand.

An example of a sensor for picking the muscular activity up – tailored to the brow muscle1 – is illustrated in fig. 2. It consists of a headband, a piezo element (in the middle of the picture), and an electronic circuit (in the little box on the right).

![Fig. 2. Headband Sensor (Optimized for the Brow Muscle)](image)

The piezo element is attached to the user’s forehead with the help of the (elastic) headband and produces a voltage potential when the user contracts the brow muscle (because

1In fact, any arbitrary muscle of choice is possible; using a different muscle just requires a suitable adaptation of the sensor.)
of the pressure against the headband) – this is called electro-
mechanical coupling. However, because of the hardware gain
induced by the electronic circuit, barely raising the eyebrow
is enough to cause a noticeable reaction. In addition, since
the sensor does not make use of conventional passive EMG
electrodes (which are usually applied for the recording of
muscular activity), it is almost insensitive to noise caused
by electro-magnetic interference – it selectively responds to
contractions of the dedicated muscle (here: the brow muscle).

As a consequence, detecting intentional contractions is easy.
Whenever the amplitude of the monitored muscular activity
exceeds a certain threshold, a contraction is detected by
definition. By decreasing or increasing this threshold, the
sensor can be made more or less sensitive, so that it eventually
really takes a conscious act to trigger the detection mechanism –
that way, even a tremor can be compensated.

The detected contractions are divided into single contrac-
tions (SC’s) and double contractions (DC’s). The first contrac-
tion after an “idle activity” is interpreted as an SC. If the user
issues another intentional contraction within a time period τ
(τ shall be called “contraction time” and usually lies between
0.5s and 1.5s, depending on the user’s preferences), the
interpretation is “upgraded” to a DC. The constant application
of this algorithm results in a (“digital”) contraction sequence.

The transformation of the time-variant input signal into that
sequence of single and double contractions is exemplified in
fig. 3. In this illustration, the amplitude of the muscular activity
is drawn at discrete points in time in the form of the signal
window in HaWCoS (which will be reintroduced further below
in the subsection on HaWCoS’ control software).

![Fig. 3. Exemplification of Contraction Sequence](image)

A very interesting (probably even surprising) property of
HaWCoS is that it exclusively focuses on this sequence in
order to enable its user to go anywhere he or she wants. The
processing principle is illustrated in fig. 4. The trick is to keep
track of an internal state, which governs wheelchair motion,
and to switch among five possible states on the basis of the
contractions issued by the user.

At the beginning, the central STOP state is active, which
also means that the wheelchair is halted. After a DC, the
system switches to the lower STRAIGHT state, and the
wheelchair is set in motion, moving forward as long as there
is no new contraction. If the next contraction eventually is a
second DC, the system reverts to STOP, and the chair is halted,
while an SC activates the LEFT state, which causes the chair
to stop moving straight and start turning left.

The rest is pretty analogous. In the LEFT state, a DC revertsto
STOP, and an SC switches to the upper STRAIGHT state
(which means that the wheelchair stops turning left and starts
moving forward again), and so on. In short: once set in motion,
the user can cycle through STRAIGHT, LEFT, STRAIGHT’,
and RIGHT with SC’s or stop the wheelchair in any state with
a DC.

A single contraction in the STOP state does not alter the
state, but otherwise has a particular meaning: as will be
explained below, the control software of HaWCoS uses an
SC in STOP to switch modes.

![Fig. 4. Transition Diagram](image)

It should be noted that this type of control exclusively
either moves the wheelchair straight or turns it – any kind of
(voluntary) curve is not supported. Moreover, the user does not
have to worry about how strong to contract the chosen muscle,
it merely has to be strong enough to exceed the threshold
(i.e., to “hit” the top edge of the signal window) – contraction
or no contraction is all that counts! All this minimizes the
amount of physical contribution requested from the user, but
it imperatively makes the control very “digital” and discrete (as
opposed to the continuous control with a conventional analog
joystick).
IV. HAWCoS Realization for MEYRA-CAN Wheelchairs

As a base wheelchair for the study reported here, the "Sprint GT" from the German manufacturer MEYRA has been used (see fig. 5). The internal control software running on that wheelchair is based on the CAN (Controller Area Network) technology, and since the HaWCoS software does not mess around with anything specific to the "Sprint" model, it works on any MEYRA-CAN wheelchair without any modifications.

The conventional control unit of MEYRA-CAN wheelchairs features an 8-pin "service" socket, which may be used by all sorts of external devices (e.g., a PC) to access the wheelchair electronics. By adhering to a strict serial protocol, the external device can set the target values for speed and direction. In fact, that is what HaWCoS does to ultimately make the wheelchair move.

A. Stand-alone Device

The stand-alone device running the HaWCoS software is depicted in fig. 6. The device is mounted on top of the standard control unit while being connected to the "service" socket mentioned above via an ordinary serial cable and to the piezo sensor via a 3.5mm plug. It gets its power supply directly from the wheelchair batteries.

The rest of the hardware is described in the following, and the software system itself is grossly introduced after that.

1) Device Hardware: The interior of the box shown in fig. 6 consists of a sophisticated micro-controller and a monochrome 128 x 64 pixel display, along with a converter transforming the 24V from the wheelchair batteries into the usual 5V supply voltage.

The display used is the EA KIT128-V24 TPB from ELECTRONIC ASSEMBLY, and the micro-controller is the ATmega128 from ATMEL. The ATmega128 offers 128K B flash memory (of which only about 22K B are currently needed), and it provides two serial ports (one to connect it to the wheelchair and one for the display) as well as an internal analog-to-digital converter (for the input signal coming from the muscle sensor). To equip the ATmega with enough processing power to sample the input data at 1kHz (while continuously updating the display output and reliably controlling the wheelchair), it is operated with a clock rate of 16MHz.

In order to modify and debug the flash program in the micro-controller, the user (or better: the developer) can plug a programming interface into the extra 9-pin socket of the box. The current version of the flash program is described next.  

2) Control Software: At the beginning, the display walks the user through the rigid startup procedure (where the order of connecting the chair and then switching it on is fixed), and it then shows a title screen which may be changed by activating one of three operational modes (see fig. 7). Each of those modes can be entered with a double contraction (DC), while a single contraction (SC) cycles to the next mode.

The two little vertical windows at the left side of this title screen are the signal window and the contraction indicator, respectively. The signal window displays a marker symbolizing the current amplitude of the input signal. Each time, this marker "hits" the upper edge of the signal window, a (single) intentional muscle contraction is detected. For making this detection more or less sensitive, the signal amplification and the "0-level" are configurable.

On detecting a single contraction the contraction indicator is "filled" and then gradually "emptied", until the "contraction time" has completely elapsed. This helps the user in keeping her or him from inadvertently producing DC's: she/he just has

4To accelerate the graphical output of the system, 64K B of pixel data have been preloaded into the display's flash memory. However, this had to be done only once and does not require any further update.

5A "hidden" fourth test mode has been included for debugging purposes and is meant for manual operation only.
to wait until the contraction indicator is completely empty, before issuing another single contraction\(^6\).

When the drive mode is entered (with a DC), the program behaves exactly as described further above (when introducing the idea of internal states): the state is changed from STOP to STRAIGHT, and the user can drive where she or he wants by issuing SC's to cycle through the STRAIGHT, LEFT, STRAIGHT', and RIGHT states (with the active state being shown on the display). After another DC, the state reverts to STOP, and the wheelchair comes to a standstill. At that point, the mode can be changed (again) if desired, but it should be noted that the state will always remain STOP in any of the other modes.

In the switch mode, the user can either modify the transition diagram by choosing direction (FORWARD/BACKWARD) and orientation (LEFT and RIGHT may be exchanged) or toggle the lighting of the wheelchair (e.g., headlight and hazard lights). To choose between several options, the user can move a selection marker by issuing SC's and finalize with a DC.

The configuration mode is quite similar to the switch mode. The user can enter values (by choosing one of a limited number of options) for such parameters as the (software) amplification of the input signal, the maximum wheelchair speed, or the "contraction time" (all with cycling SC's and a final DC).

B. PC-based Simulator

Right now, there is only one prototype of the stand-alone device. To multiply the chances of testing the system, a PC-based simulator has been implemented as a Windows\(^\text{®}\) executable (see fig. 8). Instead of the stand-alone device, any ordinary laptop computer (with Windows\(^\text{®}\) operating system) running the simulator can be used. The role of the microcontroller is in that case played by the laptop’s processor and that of the 128 × 64 display by the computer screen.

When using a laptop, the output is absolutely identical, the only differences are related to the questions how the piezo sensor and the wheelchair are connected to the computer and with what cable. Connecting the sensor is very easy: it simply plugs into the standard microphone input of the computer’s sound card, and since this usually involves a 3.5mm socket, it does not even require an adapter. For connecting the wheelchair, the laptop needs to have a serial port\(^7\). Plugging the MEYRA-specific interface cable (which features a simple level converter) between that serial port and the above mentioned "service" socket is all that is needed at this point.

In addition to perfectly mimicking the behavior of the stand-alone HaWCoS, the simulator (in combination with a laptop) can also do without the piezo-based sensor – "contractions" may then be simulated by using the laptop’s keyboard. Furthermore, the user can – if so desired – have the program simulate the wheelchair output (and wheelchair movement) graphically

\(^6\)This single contraction will then be regarded as another SC, instead of the second part of a DC.

\(^7\)However, a USB-to-serial converter also works!
Conventional Joystick

HaWCoS

90 ft.

about 63 seconds

more adequate.

superposed 0.6 ft. per second forward motion would be much orthogonal to the driving direction – if anything then the mere speed of 3 ft. per second cannot really be counted, since it is the “conventional trials”. For the “TURN states”, the turning 2 ft. per second, which has exactly corresponded to the limit in the experiment, the maximum velocity has been configured to cially true for the “STRAIGHT states” in HaWCoS: during similar limits have been chosen in both cases. This is espe-

needed to do that are presented in table I.

possible (while avoiding any wall contact!) for 40 times, i.e.,

The subject was asked to complete the course as fast as possible (while avoiding any wall contact!) for 40 times, i.e., using both input methods in 20 trials each. The averaged times needed to do that are presented in table I.

To make the two sets of trials somewhat comparable, careful attention has been paid to the maximum wheelchair speed – similar limits have been chosen in both cases. This is espe-

Normally, the subject does not rely on any alternative input strategies, but he was familiar with the muscle contraction principle used in HaWCoS nevertheless. However, even for a wheelchair driver entirely new to the system, it should be possible to master HaWCoS after a very short practice period (probably less than one day).

The subject was asked to complete the course as fast as possible (while avoiding any wall contact!) for 40 times, i.e., using both input methods in 20 trials each. The averaged times needed to do that are presented in table I.

V. EXPERIMENTAL RESULTS

In order to demonstrate the theoretical soundness as well as the practical usefulness of the HaWCoS approach, a simple 90 ft. long indoor loop has been chosen for comparing performances obtained with a conventional joystick and with HaWCoS. Only one male, 36-year-old subject – using a wheelchair because of "Friedreich Ataxia" – participated in this initial evaluation study.

Although the test results admittedly do not have any statistical significance, they are indeed impressive for that single subject. Additional tests with a larger number of subjects are planned for the near future.

As shown in the table, HaWCoS – in its present form – introduces an overhead of only about 34%, which confirms or even surpasses the result of its predecessor.

In addition to the plain numerics, one result of the experiment is the verification of the universal usability of the system. The subject indicated that driving around with the HaWCoS device was amazingly easy and straightforward.

A probably unusual property of the HaWCoS software is that it does not automatically scan through the available items (e.g., in the switch mode) at any given rate. Rather, the options are shifted in direct response to certain user actions. It is this program behavior that made the subject feel to be in total control of what is going on all the time.

Furthermore, the input method used in HaWCoS indeed turned out to be almost insensitive to external noise. In comparison to similar human-machine interface approaches, the system positions itself somewhere between EEG-based BCI's and EMG-driven devices.

On the one hand, HaWCoS requests only extremely little physical interaction, unlike many "conventional" devices relying on muscular input. On the other hand, as there is no EEG recording involved, the input processing is by far not as error-prone as in a BCI.

The subject in the experiment specifically acclaimed the system’s offering to compensate unintended (minor) contrac-

On the one hand, HaWCoS requests only extremely little physical interaction, unlike many "conventional" devices relying on muscular input. On the other hand, as there is no EEG recording involved, the input processing is by far not as error-prone as in a BCI.

Anyways, the probably most important outcome of this experiment is related to the resulting trajectory.

Fig. 9a presents the route taken when the joystick control was in use, which hardly varied over the 20 trials – its length (from point A to point B) almost perfectly matches the 90 ft. mentioned above.

This clearly contrasts to fig. 9b which displays a typical "HaWCoS route" (here, line segments correspond to "STRAIGHT states", while "TURN states" are marked with little black circles). When using HaWCoS, the shapes of the actual routes varied considerably, and they easily exceeded 100 ft. in length.

There are essentially two reasons for this. Firstly, the inherent "digital" nature of HaWCoS makes frequent direction changes (or better yet: direction corrections) indispensable. And secondly, the system has to struggle with a phenomenon that shall be called "curve problem".

Like many modern power wheelchairs, the "MEYRA Sprint" has two motorized rear wheels and two (passive) caster wheels in the front. This design makes the wheelchair navigation very flexible, e.g., by enabling its driver to turn around almost "on the spot".

While being ideal for indoor use, where there is mostly not too much room to get around without any collisions, this unfortunately renders following a perfectly straight line almost impossible (unless incorporating additional sensory information).

TABLE I

<table>
<thead>
<tr>
<th>Input Method</th>
<th>Conventional Joystick</th>
<th>HaWCoS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Route Length (Idealized)</td>
<td>90 ft.</td>
<td></td>
</tr>
<tr>
<td>Maximum Velocity</td>
<td>2 ft. per second</td>
<td>&quot;STRAIGHT states&quot;: 2 ft. per second</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&quot;TURN states&quot;: 3 ft./0.6 ft. per second</td>
</tr>
<tr>
<td>Time Needed (Averaged Over 20 Trials)</td>
<td>about 47 seconds</td>
<td>about 63 seconds</td>
</tr>
</tbody>
</table>
When the control software of HaWCoS "tells" the wheelchair to go straight, the outcome is – in most cases – not a straight line, but an (involuntary) curve, depending on the orientation of the caster wheels. This deviant behavior becomes very obvious in fig. 9b: the very first line segment (connected to point A) is perfectly straight, because the caster wheels are (by definition!) initially facing away from point B; from the first LEFT turn on, all line segments are bent toward the direction of the preceding "TURN state".

A wheelchair driver using a conventional joystick can easily compensate this dependence on the caster wheel orientation by moving the joystick in the opposite direction (to a suitable extent)\(^9\). For HaWCoS to be able to do the same thing, sensors informing the system of the orientation angle of the caster wheels would be needed.

Alternatively, the wheelchair internals might be adapted: the two rear wheels might "communicate" with each other to make sure that they rotate at the same speed (which would also require two sensors), so that STRAIGHT really means STRAIGHT!

In addition, it might be desirable to equip the wheelchair with secondary infra-red or ultra-sonic sensors and to extend the control software, so that it finally can carry out autonomous tasks, such as following walls or navigating to a user-specifiable target, while of course automatically avoiding collisions. Such an "intelligent wheelchair" (see [20], [21], [22]) would not only facilitate the application of HaWCoS – as a "digital" control method – by allowing a more "continuous" driving, it would also eliminate the "curve problem" completely (at least in certain environments).

VI. CONCLUSION

The new, extended, and enhanced version of a wheelchair control system, designed for persons with very severe physical disabilities, has been introduced. The system – called HaWCoS (for "HAnds-free Wheelchair COntrol System") – allows a wheelchair user to drive anywhere he or she wants by generating tiny contractions of an arbitrary muscle of choice – for example, it suffices to be able to intentionally raise the eyebrow.

The muscular activity of the designated muscle is transformed into a sequence of single and double contractions, which is processed to ultimately yield an internal state governing wheelchair motion. The detection of intentional muscle contractions involves a piezo-based sensor, which is (almost) insensitive to external electro-magnetic interference.

The installment of the system presented in this paper has been converted into a stand-alone device, acting as a supplementary control unit in power wheelchairs using the MEYRACAN technology. As a bonus, the system has also been implemented a simulator runnable on a standard Windows® laptop, which multiplies the opportunities to evaluate the system.

\(^9\)In doing so, the wheelchair driver can go straight forward, while actually steering a curve.
The outcome requests only a minimum amount of physical contribution from the user, but is at the same time very robust and reliable. Someone entirely new to the system can learn to master it in a surprisingly short period of time, and a simple test experiment revealed that -- in comparison to the conventional control using a manual joystick -- HaWCoS navigation requires less than 40% more time.

Unfortunately, HaWCoS also has to struggle with a number of inherent drawbacks, above all its "choppy" and discontinuous driving style: since HaWCoS only supports either going straight or turning (but not voluntarily driving curves), the resulting trajectory is affected by frequent direction changes, e.g., while the driver tries to get parallel to walls limiting the targeted route. This is aggravated by the so-called "curve problem": without additional sensory input the wheelchair used in this study is hardly able to follow a straight line, but mostly produces an involuntary curve instead, depending on the orientation of the caster wheels. A compensation also requires constant direction corrections. Therefore, HaWCoS had to lose the performance comparison with the conventional joystick.

The most eagerly demanded task for the immediate future concerns additional evaluation experiments with a statistically significant number of participants. Besides, equipping the wheelchair with a minimal set of sensors (in order to partially eliminate the inherent system problems) is something to be examined next. The ultimate goal will be to devise a product that can actually be marketed. A more advanced idea in this respect (to be considered at a later stage) is the extension of HaWCoS to an intelligent wheelchair system (which will be able to perform autonomous navigation tasks).

Finally, combining the HaWCoS simulator with the hands-free mouse control system based on the same principle (and introduced by the same authors as this paper; see [23], [24], [25]) may be a simple exercise, but it definitely is a promising job. The mentioned mouse control system empowers someone with a severe disability to operate a PC without the need to use the hands -- additionally being able to control her or his wheelchair (using the same application!) would certainly increase that person’s independence.

ACKNOWLEDGMENT

The authors would like to thank the German wheelchair manufacturer MEYRA for donating the (standard) power wheelchair used in this study and for providing ample advisory assistance.

REFERENCES