Comparison of Wheelchair User Interfaces for the Paralysed: Head-Joystick vs. Verbal Path Selection from an offered Route-Set

Christian Mandel* Udo Frese†

*University of Bremen, Department of Computer Science, P.O.Box 330440, 28334 Bremen, Germany

†DFKI Lab Bremen, Safe and Secure Cognitive Systems, Robert-Hooke-Straße 5, 28359 Bremen, Germany

Abstract— **Within this work we compare two human-robot interfaces that enable handicapped people who may not move their arms and legs to steer an automated wheelchair. As a technically more ambitious solution we first propose a simple speech recognition system that enables the paralysed to select a path from a set of spatially meaningful solutions, that is subsequently executed by a geometric path planner. The contrasting approach is given by a proportional head-joystick that evaluates the user's head posture with the help of a small-size three degrees of freedom orientation tracker. Both systems are assessed by the outcome of an empirical study in which untrained participants used the presented interfaces to navigate in a populated office environment.**

Index Terms— **Assistive Robots, Human Robot Interaction, Motion Planning, Navigation**

I. Introduction

Electrical wheelchairs give handicapped people the possibility to regain mobility in everyday life. Unfortunately, people suffering from certain spinal cord injury dysfunctions can only move body parts located above their shoulders, which leaves them unable to control a wheelchair via a common joystick interface. For this target group we propose the use of a simple speech interface with which the operator of an intelligent wheelchair selects a desired movement from an offered set of navigable routes, cf. Fig.2(a) for an illustration of potential routes in an outdoor environment. The proposed routes are derived from a frequently updated Voronoi graph that represents the immediate neighbourhood of the wheelchair. By means of a consistent denomination of spatially equivalent routes in terms of the ICAO-alphabet[1] we ensure that the operator has to command a new route only in situations where he wants to change the driving behaviour, e.g. entering a room after travelling along a corridor. The selected routes are executed by the combination of a geometric path planning module that computes obstacle-free cubic Bezier curves along with an underlying path- and velocity controller.

In order to assess the presented method, we compare it with a technically less complex approach of controlling an automated wheelchair by people suffering from paralysis of all four limbs. The system of contrast that has been first described in [12], [10] employs a three degrees of freedom orientation tracker mounted at the back of the operator's head for measuring its current posture. In analogy to a common joystick the user commands forward or backward movement by pitching his head down or up, and left or right movement by rolling his head left or right respectively.

The remainder of this paper is organized as follows. We begin in section II with an in-depth survey on human-robot interfaces, particularly focusing on the handling of autonomous wheelchairs. Section III then primarily describes our approach of controlling an autonomous wheelchair by verbal selection of proposed routes which are subsequently executed by an geometric path planning module. In the second part of this section we sketch the design of an head-joystick based steering approach for electrical wheelchairs. Both presented methods are compared in section IV by means of an experimental evaluation where we compare the participants ability to steer an autonomous wheelchair on a predefined course in a populated office environment. We conclude in section V with an assessment of the gathered results and an outlook on future work.

II. Related Work

The operation of electrical wheelchairs by disabled people through sophisticated interface techniques has its roots in the research field of human-robot interaction (HRI). In the following we give an outline on existing techniques that generally can be classified into systems that place the operator in-between the control loop of the vehicle, or those that interpret and execute abstract commands that specify more complex behaviours.

Within the first class of systems Jaffe proposes a head position interface [6]. It utilises two ultrasonic sensors that are mounted at the headrest of an electrical wheelchair in order to sense the posture of the operator's head within a two-dimensional plane parallel to the ground. The overall approach that has been evaluated within a clinical trial by 17 participants [3] maps the head's movement to proportional joystick-like signals. A comparable system that has been developed by Chen and colleagues [2] evaluates the output of two inertial tilt sensors mounted at the back of the user's head. By moving his or her head the user triggers discrete speed and steering commands, e.g. $70cm/s$ translational speed when the head is

[1] By naming routes after the alphabet of the *International Civil Aviation Organization* we try to reduce speech recognizer failures since all words out of the alphabet are selected in a way that they are easily to distinguish, e.g. no monosyllabic words that have a mutual different sound.

moved forward once or $100cm/s$ translational speed when the head is moved forward twice respectively.

Within the second class of systems mentioned above falls the work of Canzler and Kraiss [1]. The authors describe the analysis of facial features like head posture, direction of gaze, and lip movement via a sophisticated computer vision system. Therewith they show the possibility to recognize at least four gesture dependent commands like *go*, *stop*, *left* and *right* for the user of an automated wheelchair. Control via the selection of fundamental behaviours is also implemented by the general computer interface *EagleEyes* that has first been described by Gips [4] and has been ported to the application scenario of commanding the robotic wheelchair system *Wheelesley* [18].

III. Analysed Wheelchair Interfaces

A. Control Via Path-Selection from an Offered Route-Set

In the following we describe the flow of procedures necessary for controlling an autonomous wheelchair by verbally commanded routes. The involved actions can be roughly subdivided into the representation of the environment along with the computation of navigable routes, and the execution of user selected routes.

1) Voronoi-Graph as Navigation-Domain: Our experimental platform, the autonomous wheelchair *Rolland* [11] cf. Fig.1, is equipped with two laser range finders that sense nearby obstacles to the front and to the back of the vehicle in an height of about $15cm$. In combination with the information coming from two incremental encoders that measure the speeds of the two actuated wheels, the system maintains the basic representation of its environment, the so-called *Evidence Grid*. Within this square array of cells that each stores the evidence that it is occupied by an obstacle, the wheelchair is always located in the center and rotated in correspondence to its real-world heading. The current implementation of the evidence grid, that is formally defined in (1), maintains a $7.5 * 7.5m^2$ grid out of $300 * 300cells$ resulting in an spatial resolution of $2.5 * 2.5cm^2$, cf. Fig.2(b) for an illustration.

$$EG := \{egc(x,y)\} : \begin{cases} egc(x,y) \; = \; 1 \; (unoccupied) \\ \quad\vdots \qquad\qquad\quad \vdots \\ egc(x,y) \; = \; 256 \; (occupied) \end{cases} \tag{1}$$

The so-called *Distance Grid* is derived from the evidence grid and the next stride in the representation of the environment. Having the same dimensions and resolution as the evidence grid, the distance grid is calculated by a fast double sweep algorithm that computes for each free cell the metrical distance to the next occupied cell. In our current implementation of the distance grid, that is formally defined in (2), we treat a cell in the evidence grid as occupied when its value is greater than 128, cf. Fig.2(c) for an illustration.

$$\begin{gathered} DG := \{dgc(x,y)\} : \\ \forall egc(x',y') \in EG, egc(x,y) \in EG \\ \begin{cases} dgc(x,y) \; = \; 0 : & egc(x,y) > 128 \\ & \vee \; \overrightarrow{(x,y)} = \overrightarrow{(x',y')} \\ dgc(x,y) \; = \; min \mid \overrightarrow{(x,y)} - \overrightarrow{(x',y')} \mid : otherwise \end{cases} \end{gathered} \tag{2}$$

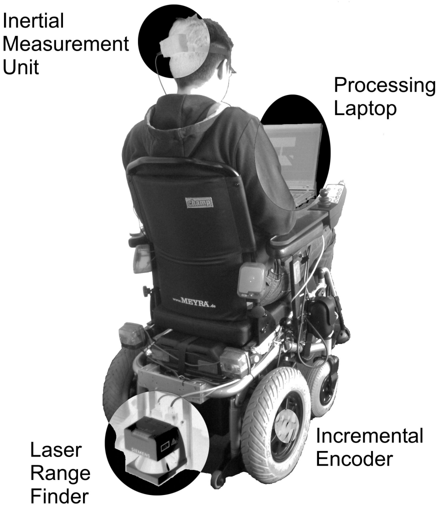
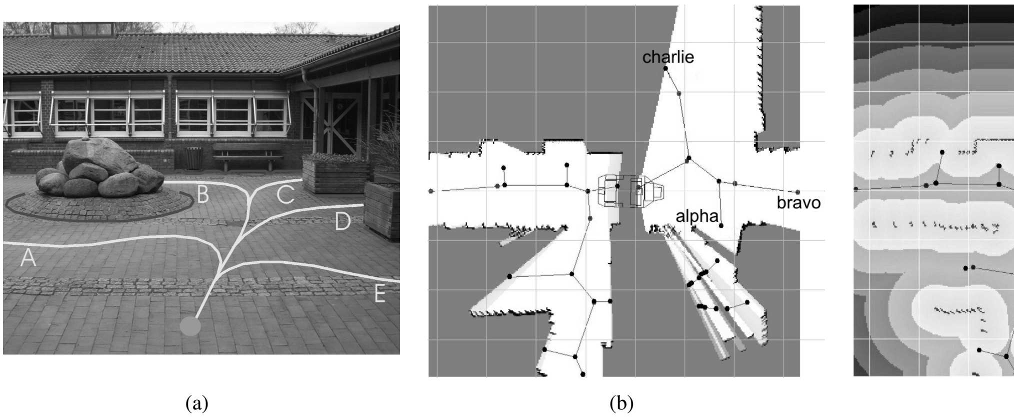


Fig. 1. The autonomous wheelchair *Rolland* along with its sensorial equipment and the processing laptop.

Computing the *Voronoi Graph* as the final stride of environmental representation requires two steps. First, we compute for each $dgc(x,y)$ out of DG whether the distance between its generating points, i.e. the occupied cells $egc(x',y')$ and $egc(x'',y'')$ out of EG that gave $dgc(x,y)$ its value, is greater than a given threshold c_d. Later on, this constant value will determine the minimal free space that is required to mark a region as navigable by an edge out of the Voronoi graph. In our current implementation we chose $c_d = 70cm$. Within the second step of computing the Voronoi graph, we search the *Voronoi Diagram* resulting from step one for points that are connected to only one or more than two neighbours. These points are finally inserted into the Voronoi graph's list of nodes V. The list of edges E consists of pairs of references to elements in V that are connected by points out of the Voronoi diagram. The Voronoi graph that is formally defined in (3) has been implemented within the general *RouteGraph*-framework [8].

$$\begin{gathered} \mathcal{VG} := (V,E) : \\ \begin{cases} VD := \{vd(x,y)\} : \mid \overrightarrow{(egc(x',y'))} - \overrightarrow{(egc(x'',y''))} \mid > c_d \\ V := \{v\} : \quad v \in VD, \quad \mid neighbours(v) \in VD \mid = 1 \\ \qquad\qquad\qquad\qquad \vee \mid neighbours(v) \in VD \mid > 2 \\ \qquad E := \{e = (v_s, v_g)\} : v_s \in V, v_g \in V \end{cases} \end{gathered} \tag{3}$$

Within the current application scenario the system graphically presents the user navigable routes that are derived from the Voronoi graph defined above. For this purpose VG is first searched by an A*-algorithm for paths connecting the vertex closest to the current odometry position, i.e. $v_{start} \in V$, to any other vertex $v_{goal} \in V$. The resulting set of paths P is then filtered for paths whose goal v_{goal} is not included in any other path $p \in P$. Within the filtered set P' we only find navigable paths that direct to spatially emphasised targets that cover desirable in-between goals by the way. For a formal



(a) (b) (c)

Fig. 2. The illustration of fundamental ideas behind the approach of commanding an autonomous wheelchair by verbal route selection from a set of offered routes ranges from Fig.2(a), the clarification of navigable routes in an unstructured environment, over Fig.2(b), the *Evidence Grid* as the basic environmental representation, up to Fig.2(c), the *Distance Grid* including the local *Voronoi Graph* along with three named routes and an cubic Bezier spline connecting the current odometry pose to vertex *charlie*.

definition of P and P' confer (4).

$$\begin{array}{lll} P & := & \{p\} : p := \{e_0, ..., e_n\}, e_0, ..., e_n \in E \\ P' & := & \{p_i\} : p_i \in P, v^i_{goal} \ni p_j \forall p_j \in P, i \neq j \end{array} \quad (4)$$

In order to offer the operator a verbally controlled selection mechanism of favoured paths, the target vertices of all paths $p' \in P'$ are now tagged by an unique name that must be easily recognisable by *Vocon*[13], the employed speech recognizer. We chose the earlier mentioned ICAO-alphabet (Alfa, Bravo, Charlie, ...) as the name-pool due to the promising properties of its elements for the recognition process. It finally remains crucial to assure that spatially equivalent paths are consistently named within consecutive processing cycles. This problem is complicated by a permanently altering Voronoi graph, caused by sensorial noise or changes in the environment and the evidence grid respectively. We tackle that issue by treating two paths as spatially equivalent only when the metrical distance between their target vertices is not greater than a given threshold. In the case of abrupt changes in the Voronoi graph that are caused by new obstacles, paths from former computation cycles may vanish and entire new paths may appear. In such cases the chosen approach stops the execution of the vanished path, and proposes the new appeared ones. This method seems to be justified since the topological change in the environment ultimately asks for a new driving command.

2) Autonomous Navigation via Bezier Curve Path Planner: The capability of our experimental platform Rolland to navigate autonomously in populated and unstructured environments is based on a geometric path planning approach that we first described in [9]. Within that work it was used to execute coarse verbal route descriptions such as: *"Leave the room, turn right, and go to the second room on the left."*. Basically, the chosen navigation method is based on the work of Hwang et al. [5], in which the authors give a general survey on navigation approaches that apply cubic Bezier curves to model obstacle-free paths within dynamic environments. The application of this kind of curve is appealing because of its spatial flexibility and the small number of determining control points $\vec{cp}_0, ..., \vec{cp}_3$, cf. (5) for a formal definition of cubic Bezier curves.

$$\begin{aligned} \vec{bc}(t) &= \vec{a}t^3 + \vec{b}t^2 + \vec{c}t + \vec{cp}_0,\ t \in [0..1] \\ \text{with } \vec{c} &= 3(\vec{cp}_1 - \vec{cp}_0), \\ \vec{b} &= 3(\vec{cp}_2 - \vec{cp}_1) - \vec{c}, \\ \vec{a} &= \vec{cp}_3 - \vec{cp}_0 - \vec{b} - \vec{c} \end{aligned} \quad (5)$$

Since the first and the last control point $\vec{cp}_0$ and $\vec{cp}_3$ determine the beginning and the end of the Bezier curve, they can easily be derived from the user-selected path $p_{sel} \in P'$ as the start-vertex v_{start} of its first edge e_0 and the goal-vertex v_{goal} of its last edge e_n respectively. The remaining control points $\vec{cp}_1$ and $\vec{cp}_2$ now span the basic search space over the cubic Bezier curves that

1) connect $\vec{cp}_0$, i.e. the current odometry position, with $\vec{cp}_3$, i.e. the user-selected goal of p_{sel},
2) are smoothly aligned with θ_s, i.e. the orientation at the current odometry position, in $\vec{cp}_0$ and with θ_g, i.e. the desired orientation in the user-selected goal of p_{sel}[2].
3) are obstacle-free in the sense that a robot-contour shifted tangentially along the curve does not intersect with any $egc(x, y) > 128 \in EG$.

In order to meet the the first and the second requirement, $\vec{cp}_1$ and $\vec{cp}_2$ are computed as points on vectors passing through $\vec{cp}_0$ and $\vec{cp}_3$, aligned to θ_s and θ_g respectively. The algebraic sign of the second summand in (6) denotes the desired direction of motion, whereas the upper sign designates forward motion and the lower one backward motion. In our current implementation we chose $l_1max = 5000mm$ and $l_2max = 5000mm$. With a fixed increment of $\delta = 75mm$, this yields an overall search space of about 4400 Bezier curves to test per frame.

$$\begin{aligned} \vec{cp}_1(l_1) &= \vec{cp}_0 \pm l_1 \begin{pmatrix} \cos(\theta_s) \\ \sin(\theta_s) \end{pmatrix},\ l_1max > l_1 > 0 \\ \vec{cp}_2(l_2) &= \vec{cp}_3 \mp l_2 \begin{pmatrix} \cos(\theta_g) \\ \sin(\theta_g) \end{pmatrix},\ l_2max > l_2 > 0 \end{aligned} \quad (6)$$

[2] The orientation in the user-selected goal of p_{sel} is given as the angle between the final edge e_n of p_{sel} and the x-axis of the odometry coordinate system.

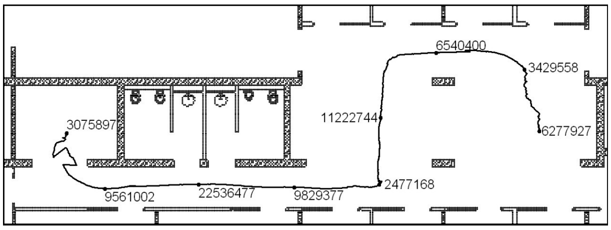


Fig. 3. The depicted trajectory is annotated with selected numbers of pixel-wise collision test operations that have been executed in order to check about 4400 Bezier curves for safety in each case.

Taking into account the third requirement that asks for obstacle-free paths, we compute for every Bezier curve out of (6) the time $t_{coll} \in [0...1]$ that indicates the first point of the curve where the shape of the robot, tangentially located on with its reference point, overlaps with an obstacle cell out of the evidence grid. Therefore any of the 4400 Bezier curves is discretized into 100 successive curve points on which we test a discretized contour of the wheelchair out of 132 contour points for collision. This yields an upper bound of about $O(58 \cdot 10^6)$ pixel-wise collision tests per computation cycle. In order to reduce the computational payload we basically skip the test of a contour cell cc' that is located nearby a contour cell cc from which we know that its minimal distance to the next obstacle is greater than the distance $\mid cc' - cc \mid$. To illustrate the result of this optimization process, Fig.3 shows an autonomously executed trajectory, as it was estimated by an adaption of the Monte Carlo localization approach used by the German RoboCup Team [15]. The driven path that corresponds to the task for several participants of our experimental evaluation phase, cf. sec. IV, is annotated by the number of completed pixel-wise collision tests. Along the approximately $25m$ long path we observed an average of $\approx 6.2 \cdot 10^6$, a minimum of ≈ 115000 and a maximum of $\approx 22 \cdot 10^6$ collision test operations per computation cycle.

We finally choose a solution of the presented path planing problem in fulfilment of requirements 1) - 3), that is obstacle-free and that minimizes the overall time of travel. The upper velocity-bound of the wheelchair at point $\vec{bc}(t)$ is therefore determined by the minimal distance between the robot-contour tangentially located at $\vec{bc}(t)$ to any obstacle-point, and the curvature $c(t)$. Once computed the final trajectory, it can be executed by the application of a path controller [14], that minimses the errors in orientation, translational velocity and lateral distance between the current wheelchair position and the closest pose on the trajectory.

B. Control Via Proportional Head-Joystick

In comparison to the preceding paragraph we will now describe our approach of controlling an automated wheelchair by interpreting the output of a miniaturized three degrees of freedom orientation tracker[3] as joystick-like signals. Therefore we will first detail the technical characteristics of the applied sensor along with its raw output data. Subsequently we characterise the algorithmic treatment of the outputted 3d-angles, i.e. their conversion into translational and rotational speed commands.

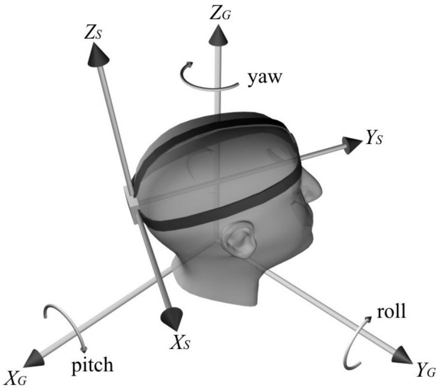


Fig. 4. The IMU that is mounted at the back of the user's head outputs drift-free *pitch*, *roll*, and *yaw* angles, i.e. the orientation of the device's local coordinate system S with respect to the fixed global coordinate system G.

1) Measuring Head Posture Angles using an IMU: The *XSens MTx* IMU [17] is a small-scale electronical device that measures $53 \cdot 38 \cdot 21mm^3$ and weights about $30g$. With its included accelerometers, gyroscopes, and magnetometers the IMU not only outputs inertial information, namely 3D acceleration, 3D rate of turn, and 3D earth-magnetic field, but also drift-free absolute 3D orientation data. Attached to the back of the user's head with the help of an easy to wear frontlet, the IMU outputs Euler angles that describe the head posture by the rotation of the IMU's local coordinate system S with respect to the fixed global coordinate system G, cf. Fig.4. A single data reading of the IMU is given as the triple $\mathcal{I} = (\psi, \varphi, \theta)$ and more precisely formulated in (7).

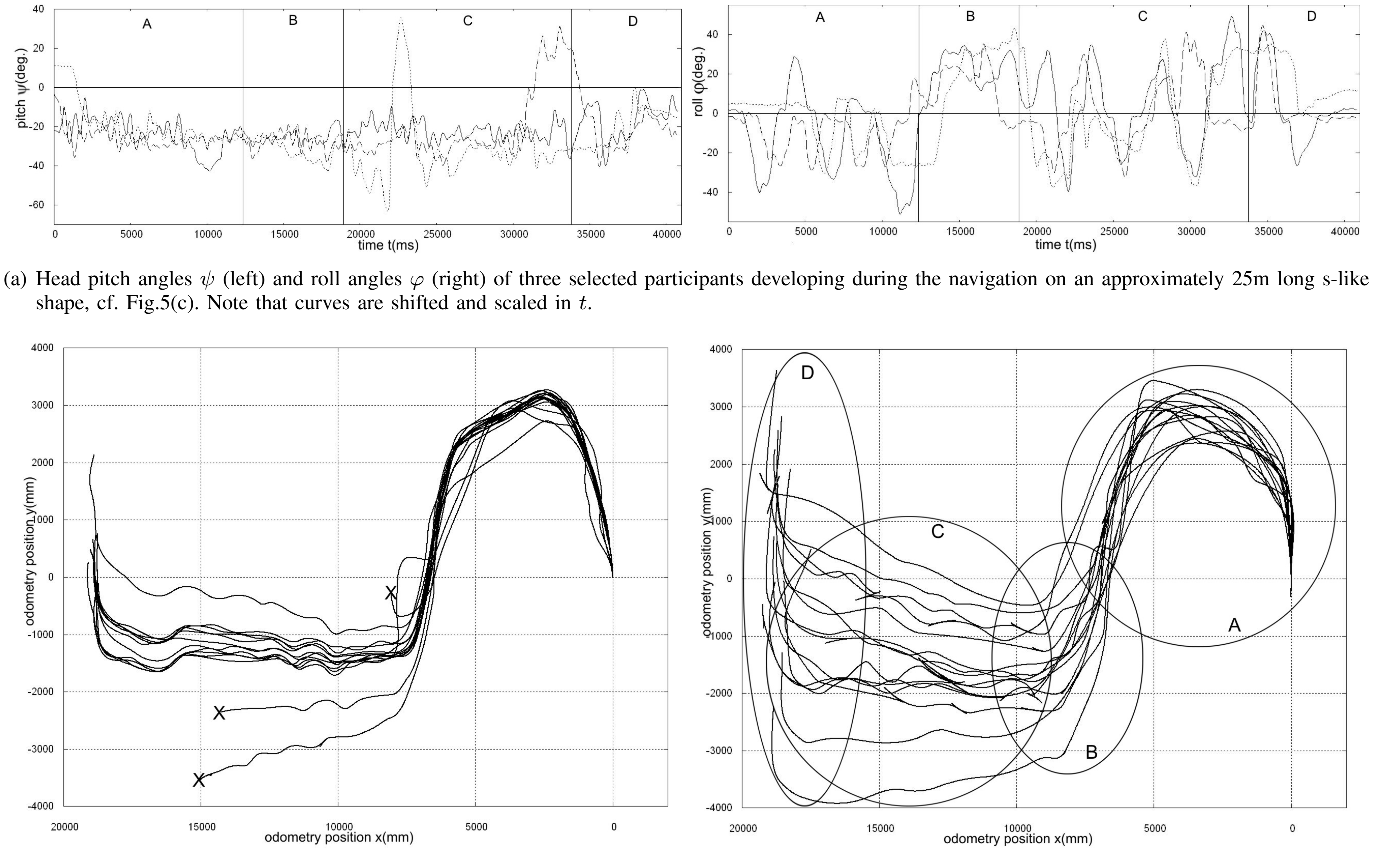
$$\begin{aligned} \psi &= pitch = rotation\,around\,X_G \in [-90°...90°] \\ \varphi &= roll = rotation\,around\,Y_G \in [-180°...180°] \\ \theta &= yaw = rotation\,around\,Z_G \in [-180°...180°] \end{aligned} \tag{7}$$

2) From Head Posture Angles to Steering Commands: The first step in the pipeline of processes that are necessary to convert a single IMU reading $\mathcal{I}$ into a pair of translational and rotational speeds $\mathcal{V} = (v, w)$ is the adoption of the minimal, maximal, and the mean deflection of the operator's head. For this purpose we ask the user to pitch and roll his or her head with maximal deflection at system start-up, while taking calibration measurements. The averaged limits ψ_{max}, ψ_{min}, φ_{max}, and φ_{min} now allow us to introduce a static dead zone around $\psi_0 = \frac{\psi_{max}+\psi_{min}}{2}$ and $\varphi_0 = \frac{\varphi_{max}+\varphi_{min}}{2}$. The dead zone as can be seen in (8) is later on used to filter unintended control commands that result from minor head movements.

$$\begin{aligned} \psi_{valid} \in \quad & [\psi_0 - c_p \mid \psi_0 - \psi_{min} \mid ...\psi_{min}] \cup \\ & [\psi_{max}...\psi_0 + c_p \mid \psi_{max} - \psi_0 \mid] \\ \varphi_{valid} \in \quad & [\varphi_0 - c_r \mid \varphi_0 - \varphi_{min} \mid ...\varphi_{min}] \cup \\ & [\varphi_{max}...\varphi_0 + c_r \mid \varphi_{max} - \varphi_0 \mid] \end{aligned} \tag{8}$$

At this stage, a valid IMU reading $\mathcal{I}_{valid} = (\psi_{valid}, \varphi_{valid})$ neglects the yaw component θ since the unconsidered rotation around Z_G allows the user to look around while driving

[3] Throughout the rest of this paper we will abbreviate *three degrees of freedom orientation tracker* or rather *inertial measurement unit* by *IMU*.



(a) Head pitch angles ψ (left) and roll angles φ (right) of three selected participants developing during the navigation on an approximately 25m long s-like shape, cf. Fig.5(c). Note that curves are shifted and scaled in t.

(b) Autonomous executed paths commanded by 5 participants using the verbal selection mechanism presented in sec. III-A. Aborted test runs are marked by a terminating X.

(c) Paths driven by 15 participants using the head-joystick from sec. III-B.

Fig. 5. The above plots show experimental data that has been gathered during an evaluation phase where 15 participants employed the proposed head joystick, cf. sec. III-B, and 5 participants the introduced verbal control scheme repeatedly, cf. sec. III-A.

his or her vehicle. In order to finally compute a velocity command $\mathcal{V} = (v, w)$ for the wheelchair, we just multiply each component of $\mathcal{I}_{valid}$ by a weighting factor that includes the reciprocal of the corresponding interval's width from (8), and a normalizing component that maps v and w onto the velocity-domain of our particular vehicle.

IV. Experimental Evaluation

The two wheelchair interfaces presented in the preceding sections have been evaluated within an empiric test series that monitored the performance of a total of 20 untrained participants to navigate in an unstructured and populated office environment. We chose a constant setting that asked every test person to trace an approximately $25m$ long reference trajectory, comprising open space navigation in our office's coffe break area, corridor following, and a final narrow entrance passage, cf. Fig.3. Because of the conceptual difference in the two applied interfaces, we will now separately present the experimental findings.

A. Evaluation of Path-Selection from Offered Route-Set

Within Fig.5(b) one can see the odometry-based paths that five participants repeatedly drove by using the presented path-selection approach from sec. III-A. A first look on the plot reveals that three out of fiveteen test runs have been aborted due to diverse failures. In the case marked by the uppermost X, the subject abandoned its attempt because it had extreme problems to interpret the map which visually displayed the offered set of routes, cf. Fig.2(b). Albeit other participants reported the same problem as less significant, it must be considered to reduce the cognitive demands in reading the related map. A possible solution is not to rotate the icon for the wheelchair w.r.t. the physical heading, but instead to rotate the whole map content and letting the contour of the wheelchair always look upwards. The aborted test runs marked by the two lower X are due to a basic shortcoming of the offered paths underlying data, i.e. the Voronoi graph based on sensorial input. In both of the two addressed cases the wheelchair stopped because of a topological change in the path tracked before. The change that was triggered by the appearing passageway at the end of the reference trajectory was however so minimal that the subsequently offered paths rapidly alternated between a straight corridor path and a branchning turn. This problem rendered the choice of a new target path impossible for the participant. Future work has to eliminate this problem by the introduction of a more stable Voronoi graph. The work of Wallgrün et al. [16] tackles that issue by restricting a given Voronoi graph to relevant vertices that are generated by major features in the evidence grid.

B. Head-Joystick Evaluation

A first impression of the participants ability to manoeuvre Rolland on the given reference trajectory by using the described head-joystick is given in Fig.5(c). The paths that together start in the middle right part of the plot are depicted in drifting odometry coordinates, whereby we can explain the strong deviations in the aimed target. One of the primary observations, path oscillations especially on the straight line navigation sector in the lower left part of the plot, can be explained by the examination of three selected test runs in Fig.5(a). The two parts show the progression of the user's head pitch angle ψ that affects translational speed, and the roll angle φ that controls rotational speed, both over time t. Analysing part C of the roll angle plot that corresponds to the straight line corridor movement, we can see that the participants controlled this section by alternating head roll movements with a direction of up to $\pm 40^\circ$. Although we believe that more experienced users are able to maintain precise straight ahead movement, a software solution in form of a so-called *Drive Assistant* would be appropriate for the unexperienced. This module would replace the actual *Safety Layer* that currently brakes in case of dangerous obstacles, by an obstacle-avoidance behaviour that alters unsafe driving commands smoothly by itself. At this point it shall be mentioned that we also observed high frequent but low amplitude oscillations of the head's roll angle φ within our first implementation. The concept of a *dynamic roll dead zone* that basically increases the clearance of unconsidered roll movements in driving situations with high translational speed eased this effect.

V. Conclusions

We have presented the implementation and experimental evaluation of two wheelchair user interfaces that allow the paralysed to control his or her vehicle without hand movements. In order to address a mutual comparison of both methods we first want to point out their conceptual difference. In [7] the authors distinguish human-robot interfaces into so-called *front-end approaches* that completely specify the task to be performed, and *incremental approaches* that decompose task descriptions into elementary actions. The head-joystick that falls into the later category comprises its main advantage of allowing for dynamic changes during the execution phase, i.e. the user is embedded in the control loop. In contrast to that, the verbal path selection approach benefits from the major advantage of the first class of interfaces in that it gives the operator maximum freedom during the autonomous execution phase.

Without paying attention to any particular user-preferences we carefully have to improve the shortcomings of both approaches. For the verbal path selection mechanism this is the already addressed stabilization of the Voronoi graph, and an improvement of the geometric path planner. The later issue primarily has to investigate the application of different types of curves in the case where Bezier curves do not adequately model specific navigation tasks, e.g. turn on the spot manoeuvres. Future work for the head-joystick interface will mainly involve long time experiments with the targeted audience from which we expect to find requirements for everyday service.

Acknowledgment

This work has been partly supported by the Deutsche Forschungsgemeinschaft (DFG) in context of the Sonderforschungsbereich/Transregio 8 *Spatial Cognition*. We also thank the Bremen members of the EU-project *SHARE-it* for their active support.

References

[1] U. Canzler and K.-F. Kraiss. Person-adaptive facial feature analysis for an advanced wheelchair user-interface. In *Proceedings of the IEEE Intl. Conf. on Mechatronics and Robotics*, 2004.

[2] Y.-L. Chen, S.-C. Chen, W.-L. Chen, and J.-F. Lin. A head orientated wheelchair for people with disabilities. *Disability and Rehabilitation*, 25(6):249–253, 2003.

[3] James M. Ford. Ultrasonic head controller for powered wheelchairs. *Journal of Rehabilitation Research and Development*, 32(3):280–284, 1995.

[4] J. Gips. On building intelligence into eagleeyes. In *Lecture Notes in AI: Assistive Technology and Artificial Intelligence*, 1998.

[5] Jung-Hoon Hwang, Ronald C. Arkin, and Dong-Soo Kwon. Mobile robots at your fingertip: Bezier curve on-line trajectory generation for supervisory control. In *Proceedings of the IEEE Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2003.

[6] David L. Jaffe. An ultrasonic head position interface for wheelchair control. *Journal of Medical Systems*, 6(4):337–342, 1982.

[7] A. Knoll, B. Hildebrandt, and J. Zhang. Instructing cooperating assembly robots through situated dialogues in natural language. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1997.

[8] B. Krieg-Brückner, U. Frese, K. Lüttich, C. Mandel, T Mossakowski, and R. Ross. Specification of an ontology for route graphs. In *Spatial Cognition IV, Lecture Notes in Artificial Intelligence*, 2004.

[9] C. Mandel, U. Frese, and T. Röfer. Robot navigation based on the mapping of coarse qualitative route descriptions to route graphs. In *Proceedings of the IEEE Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2006.

[10] C. Mandel, U. Frese, and T. Röfer. Design improvements for proportional control of autonomous wheelchairs via 3dof orientation tracker. In *Proceedings of the International Work Conference on Artificial Neural Networks (IWANN)*, 2007. To Appear.

[11] C. Mandel, K. Hübner, and T. Vierhuff. Towards an autonomous wheelchair: Cognitive aspects in service robotics. In *Proceedings of Towards Autonomous Robotic Systems (TAROS)*, 2005.

[12] C. Mandel, T. Röfer, and U. Frese. Applying a 3dof orientation tracker as a human-robot interface for autonomous wheelchairs. In *Proceedings of the IEEE Intl. Conf. on Rehabilitation Robotics (ICORR)*, 2007. Submitted.

[13] Nuance Communication Technologies. Nuance vocon speech recognizer product description and documentations, 2007.

[14] G. Oriolo, A. Luca, and M. Vendittelli. Wmr control via dynamic feedback linearization: Design, 2002.

[15] T. Röfer and M. Jüngel. Vision-based fast and reactive monte-carlo localization. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2003.

[16] Jan Oliver Wallgrün. Autonomous construction of hierarchical voronoi-based route graph representations. In Christian Freksa, Markus Knauff, Bernd Krieg-Brückner, Bernhard Nebel, and Thomas Barkowsky, editors, *Spatial Cognition IV. Reasoning, Action, Interaction: International Conference Spatial Cognition 2004*, volume 3343 of *Lecture Notes in Artificial Intelligence*, pages 413–433. Springer; Berlin; http://www.springer.de, 2005.

[17] XSens Motion Technologies. Xsens inertial measurement unit product description and documentations, 2006.

[18] H.A. Yanco. Wheelesley, a robotic wheelchair system: Indoor navigation and user interface. In *Lecture Notes in AI: Assistive Technology and Artificial Intelligence*, 1998.