

Autonomous Wheelchair for Patients with Severe Motor Disabilities

Alfredo Chávez¹, Héctor Caltenco², Kim Dremstrup¹ and Alvaro Fuentes Cabrera¹

¹*Department of Health Science and Technology, Aalborg University, Aalborg, Denmark*

²*Certec, Dept. of Design Sciences, Lund University, Lund, Sweden*

Keywords: Sensor Fusion, Autonomous Wheelchair, Laser, Occupancy Grids, Localization, Nonlinear Control, Input-Output Linearization, Path Planning.

Abstract: This paper presents the design and implementation of a control strategy for an autonomous wheelchair to assist individuals suffering from severe motor disabilities. The user is presented with a pre-generated map of a known area (e.g. home, office) displayed on a computer screen, on which the location of the wheelchair is shown. Using a specially design man-machine interface the user can select the desired point to be transported to. After the the desired point has been selected on the map, the control algorithm calculates the path and transports the user to the destination, avoiding any obstacles on its way. A Bayesian estimation method, which takes into account the uncertainty inherent in the sensor measurements, is used to fuse the sensory information obtained from a laser, and to generate and update the occupancy grid map. The proposed system uses data from a probabilistic laser map to feed a Kullback Leiber Divergence *KLD* localization algorithm and path planning based on the solution of the Laplace's equation. The system described in this manuscript is simulated in Matlab using actual measurements from a laser mounted on a mobile robot.

1 INTRODUCTION

Several electrical powered wheelchairs with modular architecture to assist mobility of disabled persons are available in the market (Edlich et al., 2004). While users with remaining hand functionality often use a joystick to control direction and speed of the wheelchair, many need different control interfaces that suit their needs. Users with severe upper-limb mobility impairments, such as spinal cord injury (*SCI*), have to rely on alternative control interfaces based on head (Christensen and Garcia, 2003), chin (Guo et al., 2002), eye (Agustin et al., 2009) or tongue (Huo and Ghovanloo, 2010; Lund et al., 2010) movements. Interfaces controlled with brain activity, commonly called brain-computer interfaces (*BCI*) (Rebsamen et al., 2007), have also been developed to assist users suffering from total motor paralysis, e.g. advanced states of amyotrophic lateral sclerosis (*ALS*). However most of these interfaces require high levels of concentration for navigating in environments with many obstacles, fixing the eyes on the screen (eye trackers) or holding the tongue in certain position for long time (tongue interfaces). Driving the wheelchair and doing activities that require the use of eyes, head or tongue at the same time might not be possible.

Another approach to augmentative mobility of disabled persons is the use of an autonomous wheelchair, which typically consists of either a standard powered wheelchair mounted with a computer and a collection of sensors or a mobile robot specially modified for the transportation of persons, (Blatt et al., 2008). The level of autonomy of the wheelchair depends on the complexity of the sensor fusion, map making, localization, path planing and control modules. The more advanced the modules are, the less interaction they should require from the user to control direction and speed movement in order to ensure a collision-free travel, e.g. avoiding obstacles, like furniture, passing through doorways, or an accurate navigation to the desired location (Tonin et al., 2010). Adding low-level autonomy to the wheelchair, while still maintaining high level control from the user, can be advantageous in many environments. Therefore, an autonomous wheelchair should require from regular to as little physical contribution as possible from the user, while still providing robust, safe and reliable mobilization.

Previous research on autonomous wheelchair has focused on different levels of autonomy, e.g., Miller and Slack (Vlassis et al., 1996) developed a wheelchair that assist the user in avoiding obsta-

cles, going to pre-designated places and maneuvering through narrow or crowded areas, making use of a variation of Dijkstra's algorithm on top of a topological map for path planing and obstacle avoidance. Civit et al (Civit-Balcells et al., 2002) presented a system that implemented recorded trajectory playback, while also providing computer assisted navigation to the user. Both systems previously described were implemented in a powered wheelchair, where users provided input using a joystick. This interaction method might be notoptimal for people with sever motor impairments. Coelho and Nunes (Coelho and Nunes, 2004) proposed an algorithm for path following control, based on a destination point paradigm, but lacks map making, path planning and localization.

In the system described in this study, the user can input the coordinates of the destination on a map of the known environment using a regular pointing device or assistive computer interface, in this paper we used an inductive wireless tongue-computer interface (Lund et al., 2010) to select the desire location on the map, previously built by the sensors. We have also included in the present study a global localization and a path planning module to generate a reference to the control input and estimate the actual pose of the wheelchair respectively. The proposed system would be able to autonomously transport the user to a desired location from a starting point in a known environment.

2 METHODS

The control algorithm described in this section has been designed to work using one input from the user, that is a destination point. When the system is turned-on the first task it performs is to display the map of the environment. Then, using the data from the laser, the system localizes the wheelchair on the map. The user is then asked to choose the desired destination on the map of the known environment, through the use of any pointing device or assistive computer interface. Then, armonic potential field path planing (Mitra, 2008; Connolly, 1997) is performed. Then, localization of the wheelchair in the previously generated map of the environment to be navigated using the Kullback Leibler Divergence (*KLD*) sampling (Hershey and Olsen, 2007). The control of the trajectory is based on input-output state feedback control, method which has been chosen due to the fact that the system has both holonomic and non-holonomic constraints, making the system not fully state feedback linearizable (Coelho and Nunes, 2003; Sarkar et al., 1994; Nijmeijer and Van der Schaft, 1990).

2.1 Sensor Fusion and Map Making

During the functioning of an autonomous wheelchair the uncertainty inherited in laser sensor data readings must be interpreted by means of a probabilistic sensor model and fused into a probabilistic occupancy map grid, where the recursive Bayesian method is used to update the probabilistic map (Elfes, 1989).

$$P^o = \frac{P_s^o P_m^o}{P_s^o P_m^o + (1 - P_s^o)(1 - P_m^o)} \quad (1)$$

where

- P_m^o and $1 - P_m^o$ are the prior probabilities that a cell is occupied and empty respectively, existing map.
- P_s^o is the conditional probability that a sensor would return the reading given the state of the cell being occupied.
- P^o is the conditional probability that a cell is occupied based on the previous sensor readings.

A new sensor reading introduces additional information about the state of the cell $C_{i,j}$. This information is obtained by the sensor model P_s^o . And, it is combined with the most recent probability estimate stored in the cell P_m^o by means of the Bayes' rule to give a new estimate P^o . The initial map's prior cell probabilities are initialized to $P_m^o = 1 - P_m^o = 0.5 \forall C_{i,j}$.

2.2 Localization within the Map

The localization of the wheelchair within the map is done using the *KLD* algorithm, which is a variant of the *MCL* algorithm in the sense that it adapts the numbers of samples over time, (Thrun, 2002).

The *KLD* implementation (algorithm 1) takes as inputs the previous sample set (X_{t-1}), the map, the most recent control odometry data (U_t) and laser measurements (Z_t). It also includes statistical error bounds ϵ and δ . In contrast, it ensures the actual sample set (X_t) and the best state estimate (x_{est}).

Lines 1 through 2 of algorithm 1 initialize X_t , and particle thresholds M and M_x . Initially each bin b in the histogram H is set to empty (lines 3 through 5). In line 7, the particle is drawn in a probabilistic way according to the weights of the particles w_{t-1}^i . The outcome is a single particle, which is then predicted, weighted and inserted in the new sample set (lines 8-10). The core of the *KLD*-sampling is implemented in lines 11 through 18. Line 11 examines whether the new particle falls into an empty bin in the histogram. If it does it, then the number of non-empty bins (k) is incremented and then the bin is set to non-empty bin.

Therefore, the number k represents the number of histogram bins filled with at least one particle. M_x gives the number of particles needed, based on the number k and the statistical error bounds ε and δ . These values are available in standard statistical tables. The algorithm main loop ends when M exceeds M_x and M_{min} . Lines 20 through 23 normalize the weights to ensure a probabilistic distribution. Lines 23 through 24 select the best state estimate x_{est} , (Thrun, 2002).

Algorithm 1: KLD-Sampling.

Require: $X_{t-1}, U_t, Z_t, map, \varepsilon, \delta$

Ensure: X_t, x_{est}

```

1:  $X_t = 0$ 
2:  $M = 0, M_x = 0, k = 0$ 
3: for all  $b$  in  $H$  do
4:    $b = \text{empty}$ 
5: end for
6: while  $M < M_x$  AND  $M < M_{min}$  do
7:   draw a particle  $i$  with a probability  $w_{t-1}^{[i]}$ 
8:    $X_t^{[M]} = \text{sample\_motion\_model}(U_t, x_{t-1}^{[i]})$ 
9:    $w_t^{[M]} = \text{measurement\_model}(z_t, x_t^{[M]}, map)$ 
10:   $X_t = X_t + \langle X_t^{[M]}, w_t^{[M]} \rangle$ 
11:  if  $x_t^{[M]}$  falls into an empty bin  $b$  then
12:     $k = k + 1$ 
13:     $b = \text{non-empty bin}$ 
14:    if  $k > 1$  then
15:       $M_x := \frac{k-1}{2\varepsilon} \left\{ 1 - \frac{2}{9(k-1)} + \sqrt{\frac{2}{9(k-1)}} z_1 - \delta \right\}^3$ 
16:    end if
17:  end if
18:   $M = M + 1$ 
19: end while
20:  $W_{total} = \sum_{i=1}^M w_i^{[i]}$ 
21: for  $i = 1$  to  $M$  do
22:    $w_i^{[i]} = \frac{w_i^{[i]}}{W_{total}}$ 
23: end for
24:  $index = \max(W_{total})$ 
25:  $x_{est} = X_t(index)$ 

```

2.3 Path Planning

To ensure free collision path planning, the occupied area (shown as white surface on the maps) has been dilated in 20cm from the contour (Gonzalez and Woods, 2002). Then, the planner finds a suitable path based on a selected destination point. Thus, for planning purposes the grid elements that represent boundary conditions like the obstacles and goal are held fixed to one and zero respectively. In the following the path planning algorithm is described.

Previous studies in robotics have used harmonic functions to solve the Laplace's equation for robot path planning (Mitra, 2008; Connolly, 1997). A harmonic function ϕ on a domain $\Omega \subset R^n$ is a function that satisfies Laplace's equation, (Connolly, 1997).

$$\Delta^2 \phi = \sum_{i=1}^n \frac{\partial^2 \phi}{\partial x_i^2} = 0 \quad (2)$$

The solution of Laplace's equation yields to equation (3).

$$\phi_{i,j} = \frac{1}{4} (\phi_{i+1,j} + \phi_{i-1,j} + \phi_{i,j+1} + \phi_{i,j-1}) \quad (3)$$

If ϕ satisfies Laplace's equation, then any grid cell $C_{i,j}$ that correspond to $\phi_{i,j}$ in the domain Ω , is the average of the cell's values at four surrounding points. This method is an iterative process that ends when there is no change of any cell grid from one iteration to the next. The algorithm creates smooth collision-free paths and does not suffer from local minima as the potential field algorithm does, (Lai et al., 2007; Connolly, 1997). 2 shows the algorithm.

Algorithm 2: Path planning using Laplace's equation.

Require: $\phi_{i,j}^0$ {The initial map};

Ensure: $\phi_{i,j}^{n+1}$ {Harmonic functions};

```

1:  $\phi_{i,j} \leftarrow 1$  {Obstacles  $(i, j)$  are fixed to 1};
2:  $\phi_{i,j} \leftarrow 0$  {Goal position  $(i, j)$  is fixed to 0};
3: while  $\phi_{i,j}^{n+1} \neq \phi_{i,j}^n$  do
4:    $\phi_{i,j}^{n+1} \leftarrow \frac{1}{4} (\phi_{i+1,j}^n + \phi_{i-1,j}^n + \phi_{i,j+1}^n + \phi_{i,j-1}^n)$ 
5: end while

```

3 MODELING AND CONTROL

Consider a nonholonomic differential drive wheelchair (front drive wheels) subjected to m constraints and n generalized coordinates (q), where $m < n$ is assumed. A geometric model is shown in Figure 1, where;

\mathcal{F}_A , fix robot frame (x_1, x_2) . \mathcal{F}_W world reference frame (x, y) . CM , center of mass. P_c , center of mass (x_c, y_c) . P_l , virtual point (x_l, y_l) . P is an intersection point between the driving wheel axis and the geometry axis of symmetry. b , distance between the center of each wheel and the geometry axis of symmetry. a , the length of the platform. d , distance from P_c to P . L , distance from P to P_l . r , radius of each wheel. w_r , right wheel. w_l , left wheel. α , the angle from CM to the center of the wheel. θ , angle of rotation.

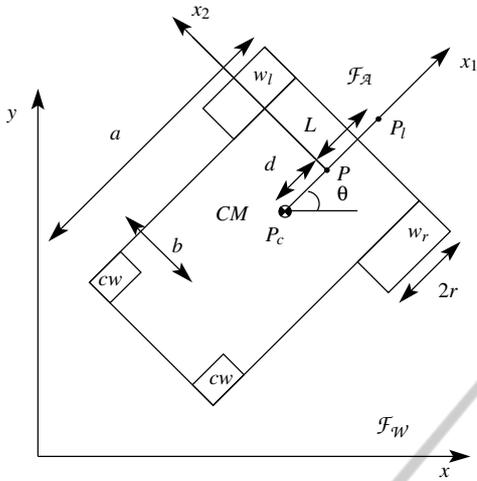


Figure 1: Differential drive wheelchair geometry. A fixed local frame \mathcal{F}_A is attached to the mobile robot and moving with respect to a world reference frame \mathcal{F}_W . θ is the angle of rotation around the center of mass CM and with respect to \mathcal{F}_W . The robot moves along x_1 and perpendicular to the driving wheel axis.

3.1 Kinematic Model

Two constraints are imposed to the kinematic model. *a*) Rolling without slipping, which means that the linear velocity of the wheel at the contact point must be zero, (Campion et al., 1996). *b*) No lateral movement constraint assumes that the wheel's orthogonal velocity components are zero, (Campion et al., 1996).

Equations (4) to (6) are the constraints imposed to the fixed wheels of the system with respect to P_c .

$$\dot{y}_c \cos\theta - \dot{x}_c \sin\theta + \dot{\theta}d = 0 \quad (4)$$

$$\dot{x}_c \cos\theta + \dot{y}_c \sin\theta + b\dot{\theta} - r\dot{\psi}_r = 0 \quad (5)$$

$$\dot{x}_c \cos\theta + \dot{y}_c \sin\theta - b\dot{\theta} - r\dot{\psi}_l = 0 \quad (6)$$

The m constraints are in the form $C(q, \dot{q})$, with k holonomic constraint and $m - k$ nonholonomic constraints, all of which can be written in the form,

$$A(q)\dot{q} = 0 \quad (7)$$

With

$$A(q) = \begin{bmatrix} -\sin\theta & \cos\theta & d & 0 & 0 \\ -\cos\theta & -\sin\theta & -b & r & 0 \\ -\cos\theta & -\sin\theta & b & 0 & r \end{bmatrix} \quad (8)$$

$$\dot{q} = [\dot{x}_c \quad \dot{y}_c \quad \dot{\theta} \quad \dot{\psi}_r \quad \dot{\psi}_l] \quad (9)$$

Some remarks about $C(q, \dot{q})$ can be stated as follows; if $C(q)$ it can be integrated, is a holonomic constraint and if $C(\dot{q})$ can not be integrated, is a nonholonomic constraint.

From the mechanical system given by equation (7), Let $S_i(q) = [s_1, \dots, s_{n-m}]^T$ be a set of smooth (continuously differentiable) and linear independent vector fields in the null space of $A(q)$, ($N(A)$), such that, $A(q)S_i(q) = 0$, $i = 1, \dots, n-m$. Hence, $S_i(q) = \dot{q}$. Now it is possible to define $(n-m)$ velocities $\eta(t) = [\eta_1, \dots, \eta_{n-m}]^T$ such that for all t , (Coelho and Nunes, 2003).

$$\dot{q} = S(q)\eta(t) \quad (10)$$

Equation (10) represents the kinematic model of a mechanical system, where $S(q)$ is a Jacobian mapping matrix from $\mathbb{R}^{(n-m)} \rightarrow \mathbb{R}^n$. In other words, it converts velocities from a mobile entity to velocities in a Cartesian system.

A Jacobian matrix $S(q)$, which satisfies the relation $A(q)S(q) = 0$, must be computed and finally a velocity vector η is stated. The computation of $N(A)$ turns out in equation (11), whereas, the velocity vector results in $\eta = [\eta_1, \eta_2]^T = [\psi_1, \psi_2]^T$.

$$S(q) = \begin{bmatrix} c(b\cos\theta + d\sin\theta) & c(b\cos\theta - d\sin\theta) \\ c(b\sin\theta - d\cos\theta) & c(b\sin\theta + d\cos\theta) \\ c & -c \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (11)$$

With $c = \frac{r}{2b}$.

3.2 Dynamic Model

The Lagrangian formalism to holonomic and non-holonomic systems has been selected to develop the dynamic equations of motion (Bloch, 2003; Goldstein, 1980).

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial L(q, \dot{q})}{\partial \dot{q}} \right) - \frac{\partial L(q, \dot{q})}{\partial q} &= M_I(q)\ddot{q} + V(q, \dot{q}) \\ &= B(q)\tau - A^T(q)\lambda \end{aligned} \quad (12)$$

$L(q, \dot{q}) = T(q, \dot{q}) - W(q)$, the Lagrangian of the system. $M_I \in \mathbb{R}^{n \times n}$, the inertia matrix. $V(q, \dot{q}) \in \mathbb{R}^{n \times n}$, the centripetal and coriolis matrix. $A^T(q)$, the Jacobian transport matrix of the constraint matrix. $B(q) \in \mathbb{R}^{n \times (n-m)}$, the input transformation matrix. $\tau \in \mathbb{R}^{(n-m)}$, an input torque vector. $\lambda \in \mathbb{R}^m$, the undetermined Lagrangian multipliers. It is assumed that the differential wheelchair is moving only in a horizontal plane, therefore the potential energy remains constant and can be neglected from the Lagrangian multiplier. The kinetic energy of a differential wheelchair with respect to P_c is stated as,

$T = \frac{1}{2}\dot{\xi}^T R^T M R \dot{\xi} + \frac{1}{2}\dot{\psi}^T I_{wm} \dot{\psi}$. With; $\dot{\xi} = [\dot{x}_c, \dot{y}_c, \dot{\theta}_c]^T$, $\dot{\psi} = [\dot{\psi}_1, \dot{\psi}_2]^T$, $I_{wm} = [I_w, 0; 0, I_w]$,

$$M = \begin{bmatrix} M_1 & 0 & M_2 \\ 0 & M_1 & M_3 \\ 0 & 0 & M_4 \end{bmatrix} R = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Where:

$$M_1 = M_2 = \frac{1}{2}(M_T + 2m), M_3 = ml \sum_{i=1}^2 \cos(\alpha_i + \theta), M_4 = \frac{1}{2}(I_T + 2ml^2 + I_w).$$

The dynamic model is obtained by replacing T into the left expression of equation (12), with; $\ddot{q} = [\ddot{x}_c, \ddot{y}_c, \ddot{\theta}, \ddot{\psi}_r, \ddot{\psi}_l]^T$, $V(q, \dot{q}) = [-2m_w d \dot{\theta}^2 \cos\theta, -2m_w d \dot{\theta}^2 \sin\theta, 0, 0, 0]^T$, $\tau = [\tau_r, \tau_l]^T$, $B(q) = [0, 0; 0, 0; 0, 0; 1, 0; 0, 1]$

$$M_I(q) = \begin{bmatrix} M_{I_1} & 0 & M_{I_2} & 0 & 0 \\ 0 & M_{I_1} & M_{I_3} & 0 & 0 \\ M_{I_4} & M_{I_5} & M_{I_6} & 0 & 0 \\ 0 & 0 & 0 & I_w & 0 \\ 0 & 0 & 0 & 0 & I_w \end{bmatrix} \quad (13)$$

Where, $M_{I_1} = M_T + 2m_w$, $M_{I_2} = -2m_w d \sin\theta$, $M_{I_3} = 2m_w d \cos\theta$, $M_{I_4} = -2m_w d \sin\theta$, $M_{I_5} = 2m_w d \cos\theta$, $M_{I_6} = I_T + 2m_w l^2 + I_w$. τ_r and τ_l , the right and left torques. M_T , the mass. $\ddot{\psi}_r$ and $\ddot{\psi}_l$, the right and left angular accelerations.

A dynamic mapping function $f: \tau \rightarrow \dot{\eta}$ is obtained by elimination of the Lagrangian multipliers from (10), where, $f_1 = S^T M_I S$ and $f_2 = S^T M_I \dot{S} \eta + s^T V$.

$$\dot{\eta} = -f_1^{-1} f_2 + f_1^{-1} \tau \quad (14)$$

Choosing the state space variable $x = [q^T \eta^T]^T$ and from (10) and (14), the kinematic and dynamic systems can be arranged into a state representation.

$$\dot{x} = \begin{bmatrix} S \eta \\ -f_1^{-1} f_2 \end{bmatrix} + \begin{bmatrix} 0 \\ f_1^{-1} \end{bmatrix} \tau \quad (15)$$

3.3 Input-Output Linearization

The system in turn is not fully input state linearizable, it may be input-output linearizable if a proper set of output equations are chosen, (Coelho and Nunes, 2003).

A state diffeomorphism transformation $z = T(x)$ and $x = T^{-1}(z)$, will bring the system in a normal form with external and internal parts respectively, making the system partially linearizable.

External and internal variables are visible and hidden to the output respectively. Moreover, a control law will bring the external part of the normal system

into a lineal one. The relative degree of the system may tell the number of outputs equations must be chosen for a specific system.

The state representation stated in equation (15) is arranged in a general nonlinear form by choosing a nonlinear feedback $\tau = f_1 u + f_2$. With; $f(x) = [S(q)\eta, 0]^T$ and $g(x) = [0, I]^T$

$$\dot{x} = f(x) + g(x)u \quad (16)$$

$$y = h(x) \quad (17)$$

Since, there are two inputs to the system $u = [u_1, u_2]^T$ two output equations must be chosen, $y = [y_1, y_2]^T = h(q) = [h_1(q), h_2(q)]^T$. For trajectory tracking control system, two output equations are chosen based on the coordinates of the virtual point P_l .

$$y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = h(q) = \begin{bmatrix} h_1(q) \\ h_2(q) \end{bmatrix} = \begin{bmatrix} x_c + (d+L)\cos\theta \\ y_c + (d+L)\sin\theta \end{bmatrix} \quad (18)$$

In order to obtain the relative degree of the system, the output y is derivate until it meets the input u .

$$\dot{y} = \Phi(q)\eta \quad (19)$$

$$\ddot{y} = \dot{\Phi}(q)\eta + \Phi(q)u$$

The output y has been derivate twice before it has encountered the input u , therefor is verified that the relative degree of the system is $\rho = 2$.

$\Phi(q)$ is the decoupling matrix and defined as $\Phi(q) = J_h(q)S(q)$ and $J_h = \frac{\partial h(q)}{\partial q} \in \mathbb{R}^{(n-m) \times n}$ as the Jacobian Matrix.

A state variable transformation vector will bring the system into an input-output linearizable one (Sarkar et al., 1994; Yun et al., 1992).

$$z = T(x) = [z_1, z_2, |z_3, z_4, |z_5]^T = [h(q), |L_f h(q), |\tilde{h}(q)]^T = [h(q), |\Phi(q)\eta, |\tilde{h}(q)]^T.$$

Where $\tilde{h}(q) \in \mathbb{R}^m$ is a vectorial function such that $[J_h^T, J_{\tilde{h}}^T]$ is a full row rank, (Sarkar et al., 1994; Yun et al., 1992). The system under the new state variable transformation vector $T(x)$ is characterized by

$$\dot{z} = [z_1, z_2, |\dot{z}_3, \dot{z}_4, |\dot{z}_5]^T = [z_3, z_4, |(\dot{\Phi}(q)\eta + \Phi(q)u), |(J_{\tilde{h}} S \eta (J_h S)^{-1}) [z_3, z_4]^T]^T.$$

The observable part of the system $\dot{z} = [z_1, z_2, z_3, z_4]^T$, is arranged as

$$\dot{z} = A_c z + B_c \Phi(q) \left[u - \left(-\Phi^{-1}(q) \dot{\Phi}(q) \eta \right) \right] \quad (20)$$

Where A_c and B_c are controllable matrices.

Choosing a state feedback control law of the form $u = \alpha(q) + \beta(q)v$, with $\alpha(q) = -\Phi^{-1}(q)\dot{\Phi}\eta$, $\beta(q) = \Phi^{-1}(q)$ and $\gamma(q) = \beta^{-1}(q) = \Phi(q)$.

$$u = -\Phi^{-1}(q)\dot{\Phi}\eta + \Phi^{-1}(q)v \quad (21)$$

Equation (21) brings the system (20) into a linear one, $\dot{z} = A_c z + B_c v$.

The kinematic and the dynamic models, the nonlinear feedback, the nonlinear state feedback control law and the linear state transformation variables are summarized in the following: $\dot{q} = S(q)\eta$, $\dot{\eta} = f_1^{-1}f_2 + f_1^{-1}\tau$, $\tau = f_1 u + f_2$, $u = \Phi^{-1}(v - \Phi\eta)$, $z_1 = h_1(q)$, $z_2 = h_2(q)$, $z_3 = \Phi_1(q)\eta$, $z_4 = \Phi_2(q)\eta$. $\Phi_1(q)$ and $\Phi_2(q)$ are the decoupling matrix members.

4 THE COMPLETE SYSTEM

The complete system, composed of the parts described in previous sections, is depicted in Figure 2. The Map making block is responsible for fusing and generating a probabilistic map of the environment based on laser readings. The path planning block has to generate a smooth path from the start to the goal. Once the path is generated, it is used as a reference, which is compared with the output from the system. The difference between the path and the reference is an error, which will be used as the input to the control. Meanwhile, the localization block together with the output from the actuators are used to generate the output signals which have to be compared with the reference input. The control block, whose dynamic equations have been discretized by means of second order Runge-Kutta method (Mathews, 1992), generates the control signals to the actuators which will follow a specific path. In order to assess the performance of the complete system, 10 simulations have been carried out in Matlab, using different starting and destination points, including transit through doorways and narrow paths, as shown in Figure 5.

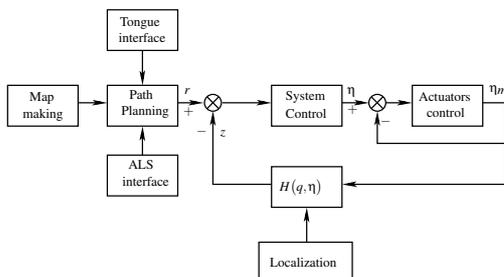


Figure 2: Block diagram of the complete system. The path planning is achieved after a destination point is selected on the map, this action produces a reference input to the control module, then the control path tracking strategy is applied to give the right commands to the actuators.

5 RESULTS

All results shown in this section correspond to the simulation of the system, performed in Matlab using data from a SICK 200 laser to build the map. A total of 30 measurements along a trajectory have been carried out by a mobile robot. In each measurement the laser scans a total of 361 readings distributed along 180° . The black area in the map represents free space C_{free} while the white area represents occupied space or C-obstacle-region CB_{region} .

5.1 Localization

For the first laser sampling, 3 iterations of the *KLD* algorithm were performed, while only one iteration was performed for the rest laser samplings as is shown in Figure 3.

- Figure 3(a) shows the result of the algorithm in the first laser measurement after one iteration, where the red circle is the true pose, the green cross is the best estimate pose calculated by the algorithm, and the blue crosses are random particles that remain after the first iteration.
- Figure 3(b) shows the result of the algorithm that is still in the first laser measurement, e.g. the mobile entity has not moved at all. The algorithm has performed three iterations, where random particles has been eliminated and the ones that remain are concentrated around the best estimate.
- Figures 3(c) and (d) shows the results for the 10^{th} and 20^{th} laser samplings after 1 iteration.

It can be seen that the *KLD* algorithm accurately tracks and localizes the mobile entity in the configuration space of the map as shown in Figure 4. Table 1 shows the true poses for different number of iterations as well as the best estimate poses in $[x, y]$ coordinates.

Table 1: Comparison between true pose and best estimate using *KLD*.

N^o Iterations	true pose	(x, y) cm Best estimate	Error
1	(225, 250)	(220, 265)	15.8
3	(225, 250)	(195, 265)	33.5
10	(585, 540)	(590.3, 527.8)	13.2
20	(540, 910)	(553.8, 908.8)	13.6

5.2 Motion Planning

Motion planning simulation based on the solution of Laplace's equation, corresponding to simulation 1 in

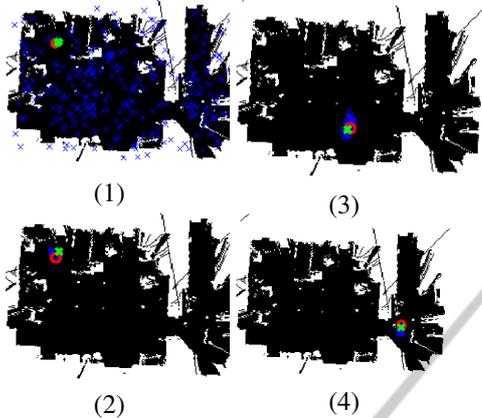


Figure 3: (a) shows the result of the algorithm for the first laser sampling after 1 iteration. (b) result of the algorithm for the first laser sampling after 3 iterations. (c) result of the algorithm for the 10th laser sampling after 1 iterations. (d) result of the algorithm for the 20th iteration after 1 iterations.

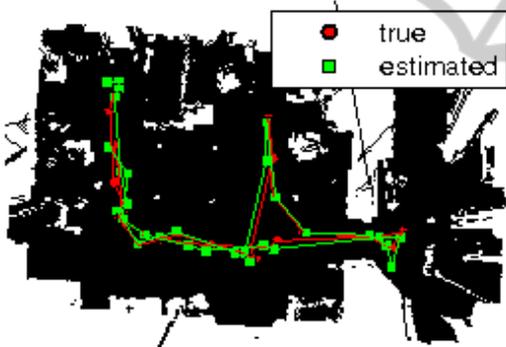


Figure 4: Comparison between the true poses and best estimates.

table 2, is shown in Figure 5. It also depicts a start point (blue circle) and a destination point (red square).

5.3 Control Simulation

Input-output state feedback nonlinear control path following for a cosine reference path is depicted in Figure 6. Where, y_1, y_2 represents the coordinates of the virtual point P_l and x_c, y_c represents the coordinates of the center of mass.

5.4 System Simulation

Table 2 shows the Mean Square Error between the reference input and the estimation of the localization

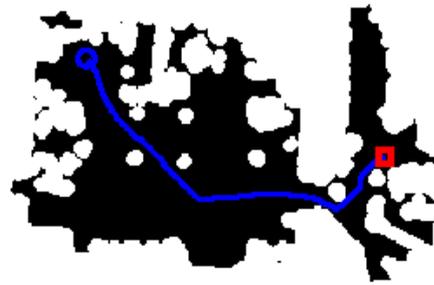


Figure 5: Configuration space created by dilation of the occupied spaces. The blue circle represents the start point configuration. The red square represents the the goal configuration coordinates chosen either by a tongue or a BCI interfaces. A smooth path (τ) in the free configuration space (C_{free}) is shown.

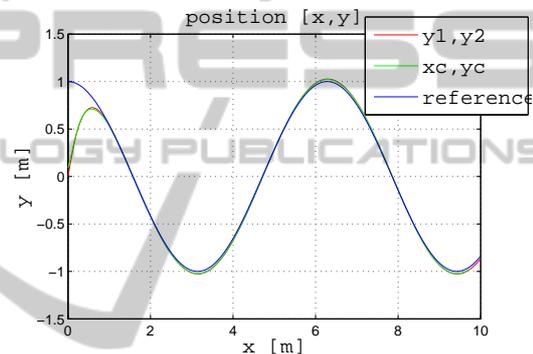


Figure 6: Shows the path following control strategy for a cosine path. The robot pose is at the origin of the coordinate system, then it reaches the path reference and follow it.

algorithm ($MSErc$), and the Mean Square Error between the reference input and the control tracking ($MSEre$) for the 10 simulations performed in Matlab. The mean and standard deviation of $MSErc$ and $MSEre$ are displayed in the bottom rows.

Figure 7 shows the nonlinear control path corresponding to simulation 1. The reference input is represented as a solid line, and the output from the control algorithm is represented as a dash line. Figure 8 depicts the estimation localization tracking corresponding to simulation 1 in table 2. The reference input is represented as a solid line, whereas the coordinates from the localization algorithm are represented as a dashed line.

6 CONCLUSIONS

The simulation of an autonomous wheelchair has been described in the present study, including modules for sensor fusion, map making, localization, path

Table 2: Mean Square Error between the reference input and estimation of the localization algorithm ($MSErc$), and between the reference input and the control tracking ($MSEre$) for all 10 trajectory simulations described in .

Simulation	$MSErc [cm^2]$	$MSEre [cm^2]$
1	6	11
2	3.7	13.1
3	3.8	5.1
4	3.9	5.1
5	5.7	8.6
6	11.5	28.1
7	6.5	21.7
8	5.2	3.2
9	6	8.2
10	5.5	8.5
Mean	5.8	11.2
Std	2.2	7.9

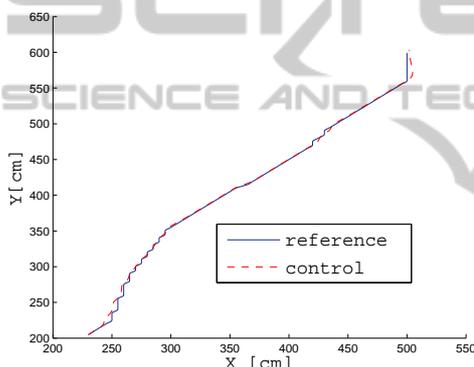


Figure 7: Control simulation, the reference input is represented as a solid line, the control trajectory tracking is represented as a dash line.

planing and control strategy.

The *KLD* localization algorithm considerably reduced the number of particles during the localization tracking after 3 iterations (Figure 4) . In table 1 can be observed that the best pose estimation for the first laser sampling, which represent the wheelchair in still position (initialization), has an accuracy error of 15.8cm and 33.5cm for 1 and 3 iterations respectively. For the 10th and 20th laser samplings, which represents the position of the wheelchair, the accuracy errors are 13.2 and 13.6 cm respectively. The errors corresponding to the wheelchair in movement are smaller than the dilation of the occupied spaces (20cm), which ensures a free collision trayjectory.

Statistical analysis of ten trajectory simulations of the system showed mean squared errors of 11.5cm² and 28cm² for localization $MSErc$ and tracking $MSEre$ respectively, what ensures accurate navigation due to the 20cm dilation of the occupied spaces.

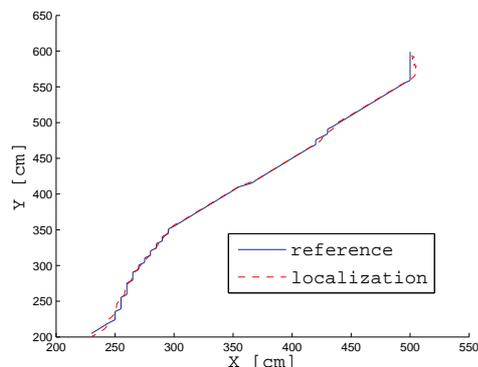


Figure 8: Estimation simulation, the reference input is represented as a solid line, the estimated localization trajectory tracking is represented as a dash line.

Results have been presented from a computer simulation of an autonomous differential drive wheelchair, using its kinematic and dynamic models. Accurate control of the real wheelchair would require attentions for more practical issues, e.g. non-holonomic constraints, such as for instance the influence of tyres etc. Stability of internal variables can be analyzed by zero dynamics of the system, (Khalil and Grizzle, 2002). This would be specially useful after having added the real-wheelchair non-holonomic constraints and other necessary control parameters. However, the simulation results in this paper have shown the feasibility of using the proposed system for an autonomous wheelchair control.

Several interfaces exist that provide users with severe mobility impairments the possibility to control a wheelchair, e.g., using the chin, mouth, tongue, eyes or even brainwaves. However most of these interfaces require high levels of concentration, or interfere with other parallel activities that could be done while driving the wheelchire, such as looking around the environment (gaze trackers, head or chin joysticks), talking (tongue-interfaces), or even getting distracted (brain-computer interfaces). Therefore, driving the wheelchair and doing activities that require the use of eyes, head or tongue at the same time might difficult. This work, is the first step for a combination of low-level autonomous control, and high-level user input, based on a single command from the user, to provide augmentative mobilization to individuals with severe motor impairments, such as high-level spinal cord injury or advanced states of amyotrophic lateral sclerosis. Such persons could use e.g. the *ITCS*(Andreasen Struijk et al., 2009) or a *BCI* (Cabrera and Dremstrup, 2008), in order to successfully choose destination points and override autonomous driving when needed.

REFERENCES

- Agustin, J. S., Mateo, J. C., Hansen, J. P., and Villanueva, A. (2009). Evaluation of the potential of gaze input for game interaction. *PsychNology Journal*, 7(2):213–236.
- Andreasen Struijk, L. N. S., Lontis, E. R., Bentsen, B., Christensen, H. V., Caltenco Arciniega, H. A., and Lund, M. E. (2009). Fully integrated wireless inductive tongue computer interface for disabled people. In *Proceedings of the 31st Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 547–550.
- Blatt, R., Ceriani, S., Dal Seno, B., Fontana, G., Matteucci, M., and Migliore, D. (2008). Brain control of a smart wheelchair. In *10th International Conference on Intelligent Autonomous Systems*.
- Bloch, A. (2003). *Nonholonomic mechanics and control*. Springer Verlag.
- Cabrera, A. and Dremstrup, K. (2008). Steady-State Visual Evoked Potentials to drive a Brain Computer Interface. Technical report, Aalborg University. Department of Health Science and Technology Aalborg.
- Campion, G., Bastin, G., and Dandrea-Novél, B. (1996). Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. *Robotics and Automation, IEEE Transactions on*, 12(1):47–62.
- Christensen, H. and Garcia, J. (2003). Infrared Non-Contact Head Sensor, for Control of Wheelchair Movements. *Assistive Technology: From Virtuality to Reality, A. Pruski and H. Knops (Eds) IOS Press*, pages 336–340.
- Civit-Balcells, A., Diaz Del Rio, F., Jimenez, G., Sevillano, J., Amaya, C., and Vicente, S. (2002). SIRIUS: Improving the maneuverability of powered wheelchairs. In *Control Applications, 2002. Proceedings of the 2002 International Conference on*, volume 2, pages 790–795. IEEE.
- Coelho, P. and Nunes, U. (2003). Lie algebra application to mobile robot control: a tutorial. *Robotica*, 21(05):483–493.
- Coelho, P. and Nunes, U. (2004). Path following control of a robotic wheelchair. *IAV2004-PREPRINTS 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles Instituto Superior Técnico, Lisboa, Portugal July 5-7, 2004*.
- Connolly, C. (1997). Harmonic functions and collision probabilities. *The International Journal of Robotics Research*, 16(4):497.
- Edlich, R., Nelson, K., Foley, M., Buschbacher, R., Long, W., and Ma, E. (2004). Technological advances in powered wheelchairs. *Journal of long-term effects of medical implants*, 14(2):107.
- Elfes, A. (1989). Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57.
- Goldstein, H. (1980). *Classical mechanics*. Addison-Wesley.
- Gonzalez, R. and Woods, R. (2002). *Digital image processing*. Prentice Hall.
- Guo, S., Cooper, R., Boninger, M., Kwarcia, A., and Ammer, B. (2002). Development of power wheelchair chin-operated force-sensing joystick. In *[Engineering in Medicine and Biology, 2002. 24th Annual Conference and the Annual Fall Meeting of the Biomedical Engineering Society] EMBS/BMES Conference, 2002. Proceedings of the Second Joint*, volume 3, pages 2373–2374. IEEE.
- Hershey, J. and Olsen, P. (2007). Approximating the kullback leibler divergence between gaussian mixture models. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 4, pages IV–317–IV–320.
- Huo, X. and Ghovanloo, M. (2010). Evaluation of a wireless wearable tongue-computer interface by individuals with high-level spinal cord injuries. *Journal of Neural Engineering*, 7(2).
- Khalil, H. and Grizzle, J. (2002). *Nonlinear systems*, volume 3. Prentice hall New Jersey.
- Lai, L., Wu, C., and Shiue, Y. (2007). A potential field method for robot motion planning in unknown environments. *JOURNAL-CHINESE INSTITUTE OF ENGINEERS*, 30(3):369.
- Lund, M. E., Christensen, H. V., Caltenco Arciniega, H. A., Lontis, E. R., Bentsen, B., and Andreasen Struijk, L. N. S. (2010). Inductive tongue control of powered wheelchairs. In *Proceedings of the 32nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 3361–3364.
- Mathews, J. (1992). *Numerical methods for mathematics, science, and engineering*. Prentice Hall.
- Mitra, A. (2008). Finite Difference Method for the Solution of Laplace Equation. *Department of Aerospace Engineering, Iowa State University*.
- Nijmeijer, H. and Van der Schaft, A. (1990). *Nonlinear dynamical control systems*. Springer.
- Rebsamen, B., Burdet, E., Guan, C., Zhang, I. H., Teo, C. L., Zeng, Q., Laugier, C., and Ang, Jr., M. H. (2007). Controlling a wheelchair indoors using thought. *IEEE INTELLIGENT SYSTEMS*, 22(2):18–24.
- Sarkar, N. et al. (1994). Control of mechanical systems with rolling constraints. *The International Journal of Robotics Research*, 13(1):55.
- Thrun, S. (2002). Probabilistic robotics. *Communications of the ACM*, 45(3):52–57.
- Tonin, L., Leeb, R., Tavella, M., and Perdakis, S. (2010). The role of shared-control in BCI-based telepresence. In *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on*, pages 1462–1466. IEEE.
- Vlassis, N., Sgouros, N., Efthivoulidis, G., Papakonstantinou, G., and Tsanakas, P. (1996). Global path planning for autonomous qualitative navigation. In *ictai*, page 354. Published by the IEEE Computer Society.
- Yun, X., Kumar, V., Sarkar, N., and Paljug, E. (1992). Control of multiple arms with rolling constraints. In *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*, pages 2193–2198. IEEE.