High Performance Control for Graceful Motion of an Intelligent Wheelchair

Shilpa Gulati and Benjamin Kuipers
Departments of Mechanical Engineering (SG) and Computer Sciences (BK)
The University of Texas at Austin
Email: gulati@mail.utexas.edu, kuipers@cs.utexas.edu

***Abstract*— To be acceptable to human drivers, the motion of an intelligent robotic wheelchair must be more than just collision-free: it must be graceful. We define graceful motion as being safe, comfortable, fast and intuitive.**

In this paper, we quantify these properties of graceful motion, providing formal evaluation criteria. We propose a method for graceful motion and present implementation results for the task of driving through a narrow doorway, evaluated on a simulated model of the wheelchair.

We use B-splines to specify an intuitive path to a goal, and then describe path-following control law for a differential-drive wheeled vehicle to follow that path within velocity and acceleration bounds.

Existing methods typically respond to tight clearances with very slow motion which is not graceful. Our results show that, starting from a set of representative poses, the wheelchair passes through the door at near maximum speed, staying close to the mid-line of the doorway. The velocity of the wheelchair reflects the curvature of the path rather than the closeness of the door edges, so it can move smoothly, safely, and quickly through the doorway.

Thus, this paper makes two contributions - first it introduces the concept of graceful motion and provides quantitative measures for the same, and second, it proposes a method for graceful motion and demonstrates it on a specific task.

I. Introduction

An intelligent wheelchair that navigates safely through the environment without requiring constant control from its human driver can reduce driving-stress and increase mobility of a disabled person [1], [2]. The ability of the wheelchair to gracefully perform common navigation tasks such as going through a door, going up and down ramps, and turning sharp corners is an important factor in ensuring the comfort of the human driver.

To be *graceful*, motion must be visibly safe, comfortable, fast, and intuitive. Safety means that motion is collision free, maintains sufficient clearances from obstacles and has low jerk. Jerk is rate of change of acceleration and a high jerk implies that the forces acting on the wheelchair have changed rapidly. High jerk can not only make the motion uncomfortable but also unsafe for people with specific conditions such as spinal cord injuries.

Comfort means that the wheelchair maintains sufficient clearances from its surroundings, its velocity and accelerations are smooth and bounded and do not oscillate, and it has low jerk. While some bounds are necessarily imposed by physical limitations of motion and by wheelchair hardware, others can be determined by the driver. For example, some people might consider lower velocities than that prescribed by the above limitations as comfortable and prefer to set lower velocity bounds.

Fast motion means that the wheelchair moves as fast as permitted by its velocity bounds and the curvature of the path. Thus, when the wheelchair passes through spaces with tight clearances such as doorways, it does not move slowly because it is between two obstacles. Rather, it moves at near maximum speed allowed by the curvature of the path.

Intuitive motion means that the paths taken by the wheelchair appear natural to the driver. The notion of a natural path is subjective: for example, consider the task of driving a wheelchair through a narrow doorway when the wheelchair is facing away from the door. While one person may like to first turn to face the door and then drive toward it, another may prefer to start driving and turn while moving. In all cases, the path should be so defined that it allows the criteria for safety and comfort to be met. For example, a discontinuous curvature curve does not allow smoothly varying angular velocity and cannot be used to specify an intuitive path.

While a significant amount of work has been done on mobile robot navigation, little attention has been paid to graceful navigation. Navigation methods that have been primarily developed for collision avoidance include the dynamic window approach [3], the curvature-velocity method [4] and the nearness diagram method [5]. These methods compute velocities at each time step to optimize an objective function that embodies various performance measures such as distance from obstacles and progress toward the goal.

These methods are completely reactive: the velocities chosen at each time step determine an arc that the robot will follow in that time step. Since there is no look-ahead, the velocities are decided based only on the current time step. This makes it impossible for the robot to reach a goal that requires a drastic change in velocities in the next step without violating the acceleration constraints. Reactive methods typically reduce the velocity of the robot when approaching an obstacle, so motion through a space with tight clearances, such as a doorway, is very slow.

Like reactive methods, safety is our most important concern. Our method is to plan a smooth collision-free route with low curvature in the tightest part allowing the robot

to navigate tightly constrained spaces as fast as comfortably possible (since lower curvature implies higher velocity), and then use a control law to drive the robot on that path while obeying the velocity and acceleration bounds. Using a pre-planned path allows the robot to take advantage of the knowledge of the future to adjust its velocities in the present. For example, once it is known that the path will turn sharply in the future, the robot can slow down in anticipation and make the sharp turn comfortably. In this paper, we use splines to specify an intuitive path. The paths are described by manually specifying control points for the spline curves and no planning algorithms are used since our purpose here is to demonstrate the ability to follow a given path gracefully. However, there exists a vast body of literature on robot motion planning [6], [7], [8], [9], some of which can be adapted to generate collision-free paths whose curvatures have continuous first and second derivatives so that it is possible to move gracefully along these paths.

II. Path Following Control Law

A path following control law was first proposed in [10] and later improved upon in [11]. The proposed law is guaranteed to converge to a given path and produces smooth motion. However, this law does not impose any bounds on velocities or accelerations. We build upon this law by formulating a constrained optimization problem to determine the parameters used in the control law such that the resulting velocities are smooth and vary smoothly, resulting in graceful motion.

A path following control law has two objectives: first, it should reduce the distance between the robot and the path to zero, and second, it should drive the angle between the robot's forward velocity vector and the tangent to the path to zero. The notion of "distance from the path" is not well-defined here. The simplest definition would be to define the distance as the perpendicular distance from the robot to the path. However, this results in singularities in the kinematic equations when the distance of the robot from the path is equal to the radius of curvature [10].

To overcome this difficulty, "distance from the path" can be defined as the distance of the robot from a virtual target on the path and virtual target dynamics can be specified so that it is not the nearest point to the robot on the path [11].

A. Kinematic Model of the Intelligent Wheelchair

The intelligent wheelchair consists of two independently driven parallel rear wheels, two free castor wheels at the front and one free castor wheel at the back for stability. Let (x, y) be the midpoint of the two rear wheels and θ_m be the orientation of its main axis in a world frame as shown in Figure 1.

The kinematic model of the wheelchair is the unicycle model and in the world frame $\{\mathrm{W}\}$ it is given by

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta}_m \end{pmatrix} = \begin{pmatrix} v\cos\theta_m \\ v\sin\theta_m \\ \omega_m \end{pmatrix} \tag{1}$$

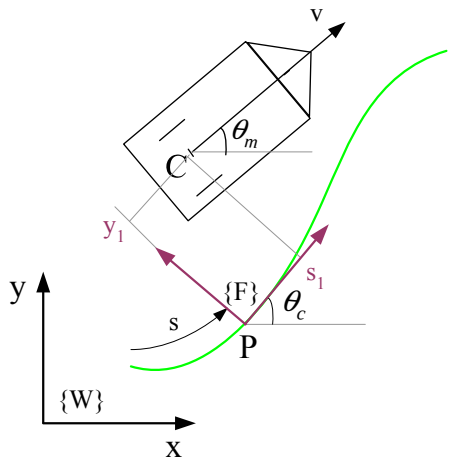


Fig. 1. The World frame $\{\mathrm{W}\}$, Serret-Frenet frame $\{\mathrm{F}\}$ and the unicycle model

where v and ω_m are the linear and angular velocities in the world frame respectively.

For deriving the path following control law, it is useful to think of the kinematic model of the robot in the Serret-Frenet frame $\{F\}$ attached to the virtual target as shown in Figure 1. The position of the virtual target $(P_x(s), P_y(s))$ is specified by the arc-length s along the curve. The x axis of the frame is along the tangent to the path at that point. Let the orientation of the robot in the Serret-Frenet frame be θ. Then, the pose of the robot in this frame is (s_1, y_1, θ). Each of these coordinates represents the error that must be reduced to zero.

B. Kinematic Model in the Serret-Frenet Frame

Let θ_c be the angle of the tangent vector to the path at s in the world frame $\{\mathrm{W}\}$. Let θ_m be the orientation of the robot in the world frame. Then, the angle θ in the Serret-Frenet frame is given by

$$\theta = \theta_m - \theta_c \tag{2}$$

We would like to define θ such that it is the smallest angle between the robot's axis and the path, hence we bound θ to $[-\pi, \pi]$. Let c_s be the curvature of the path at arc-length s. Then, we have

$$\dot{\theta}_c = c_s \dot{s} \tag{3}$$

The kinematic model of the robot in the Serret-Frenet frame is [11]

$$\begin{aligned} \dot{s_1} &= -\dot{s}(1 - c_s y_1) + v\cos\theta \\ \dot{y_1} &= -c_s \dot{s} s_1 + v\sin\theta \\ \dot{\theta} &= \omega_m - c_s \dot{s} \end{aligned} \tag{4}$$

C. Control Law using Kinematic Model

It has been shown in [11] that the robot converges to the path if $\dot{s}$ and $\dot{\theta}$ are defined as

$$\begin{aligned} \dot{s} &= v\cos\theta + k_1 s_1 \\ \dot{\theta} &= \dot{\delta} - \gamma y_1 v \frac{\sin\theta - \sin\delta}{\theta - \delta} - k_2(\theta - \delta) \end{aligned} \tag{5}$$

where k_1 and k_2 are positive constants, and the shaping function $\delta(y_1, v)$ satisfies

1) $\delta(0, v) = 0$
2) $y_1 v \sin(\delta(y_1, v)) \leq 0 \quad \forall y \quad \forall v$
3) $\lim_{t\to\infty} v(t) \neq 0$.

Condition (1) states that the shaping function is zero when the robot is on the path ($y_1 = 0$), so the robot's main axis must be tangent to the path. Condition (2) adjusts the sign of the shaping function such that the robot always turns toward the path, no matter which side of the path it is on and what direction it is moving in. For example, for positive v, if y_1 is negative, condition (2) implies that $\sin(\delta(y_1, v))$ is non-negative and hence $\delta(y_1, v) \in [0, \pi]$. Condition (3) ensures that the robot does not come to a stop.

D. A Shaping Function

We use a modified version of the shaping function proposed in Lapierre *et al.* [11].

$$\delta = -\theta_a \tanh(k_\delta y_1) \tag{6}$$

under the assumption that $v \geq 0$ and $0 \leq \theta_a \leq \pi$, and k_δ is a positive constant. Differentiating Equation 6, we get

$$\dot{\delta} = -\theta_a k_\delta \dot{y_1}(1 - \tanh^2(k_\delta y_1)) \tag{7}$$

E. A Closer Look at the Control Law

From Equations 2, 5 and 7 we get the control law for angular velocity as

$$\omega_m = -\theta_a k_\delta \dot{y_1}(1 - \tanh^2(k_\delta y_1)) - \gamma y_1 v \frac{\sin\theta - \sin\delta}{\theta - \delta} - k_2(\theta - \delta) + c_s \dot{s} \tag{8}$$

To understand the control law, assume that the constant k_1 in Equations 5 is zero and that the linear velocity follows a given profile $v_d(t)$. Suppose that the robot is on the path ($s_1 = 0$, $y_1 = 0$) and is tangent to the path such that $\theta = 0$. In this case, only the last term is non-zero and the law reduces to $\omega_m = c_s v$. This merely adjusts the angular velocity such that the robot stays on the path. Next, suppose that the robot is on the path ($s_1 = 0$, $y_1 = 0$) but not tangent to the path ($\theta \neq 0$). The first term is now a function of θ (substitute $\dot{y_1}$ from equations 4) and the second term is zero. The first and third terms rotate the robot so that it becomes tangent to the path while the fourth term makes it move forward along or parallel to the path.

Now, suppose that the robot is not on the path but parallel to it such that $\theta = 0$. The third term is now zero. If the first two terms were not present, the robot will merely move along a curve parallel to the path as there is nothing pushing it back to the path. These terms ensure that the robot moves back to the path. Finally, when the robot is not on the path, all the terms serve to make it converge to the path.

III. Control Law and Paths for Graceful Motion

The control law described in the previous section assumes that the linear velocity v follows a desired velocity profile $v_d(t)$ and imposes no bounds on velocities and accelerations. Notice that in Equation 8, we have the freedom to choose two parameters, namely the linear velocity v and the constant k_2, provided that both of them are positive. To determine these parameters at each time step we formulate an optimization problem where the objective is to optimize a function $f(v, k_2)$ subject to constraints imposed by bounds on velocities and accelerations.

A. Formulating the Optimization Problem

Assume that initially the robot is on the path and tangent to the path. Define an optimization problem to maximize the linear velocity v as follows

$$\begin{aligned} &\text{Maximize} \\ &\qquad f(v, k_2) = v \\ &\text{subject to} \\ &\qquad \epsilon \leq k_2 \\ &\qquad 0 \leq v \leq v_{max}, \\ &\qquad a_{min}\triangle t + \hat{v} \leq v \leq a_{max}\triangle t + \hat{v} \\ &\qquad \omega_{min} \leq \omega_m \leq \omega_{max} \\ &\qquad \alpha_{min}\triangle t + \hat{\omega}_m \leq \omega_m \leq \alpha_{max}\triangle t + \hat{\omega}_m \\ &\qquad \alpha_{min} \leq c_s \dot{v} + g_s v^2 \leq \alpha_{max} \end{aligned} \tag{9}$$

where ω_m is given by Equation 8 and can be written as a function of v and k_2, given values of s_1, y_1, θ and the constants k_1, k_δ, θ_a and γ (using Equations 4). The quantities with a hat are the values from the previous time step, v_{max}, ω_{max}, a_{max} and α_{max} are constants that represent the maximum permissible values of linear velocity, angular velocity, linear acceleration and angular acceleration respectively (similarly the minimum permissible values), g_s is the derivative of the curvature with respect to s, and ϵ is a small positive constant.

The first constraint represents the fact that k_2 is a positive constant. The next four constraints represent the bounds on velocities and accelerations and do not take into account the fact that the curvature of the path may be changing. These constraints are local and unable to change the velocities in anticipation of an impending change in curvature.

The final constraint takes into account the rate of change of curvature and thus introduces a kind of look-ahead in the system, allowing it to reduce its linear velocity before an impending sharp turn or to speed-up if the curvature is decreasing. This constraint is an approximation, it is derived assuming that the robot is on the path at that time step. Then, as explained in the previous section, the control law becomes

$$\omega_m = c_s v \tag{10}$$

Taking the derivative we get

$$\dot{\omega}_m = c_s \dot{v} + g_s v^2 \tag{11}$$

Imposing bounds on $\dot{\omega}_m$ then gives us the final constraint.

B. Defining Paths as Spline Curves

We discussed a control law that allows the robot to track any given path in two dimensions with bounded velocities and accelerations, provided its arc-length parametrization is available. An easy and computationally cheap way of specifying a path in two and three dimensional space is by defining a spline curve. Splines allow lower degree curves as compared to polynomial interpolation to be fitted to data, and have been widely used in motion planning and computer graphics to specify smooth paths [12], [13]. We choose a particular class of splines called non periodic B-splines for specifying paths.

A B-spline is a piecewise polynomial curve that approximate a set of points. It is a parametric curve of the form [14]

$$\mathbf{p}(u) = \sum_{i=0}^{n} \mathbf{p}_i \; N_{i,k}(u) \tag{12}$$

where u is the parameter, $\mathbf{p}_i$ are the control points, $N_{i,k}$ are the blending functions, and $k-1$ is the degree of each piece of the curve. The control points determine the characteristic polygon of the curve. Each vertex $\mathbf{p}_i$ of the characteristic polynomial is weighted by a blending function $N_{i,k}$. The piecewise curves can be up to C^{k-2} continuous at the joints.

The blending functions for a non periodic B-spline are defined such that for any value of the parameter u, all but k blending functions are zero. Thus, the shape of any segment of the curve is determined by only k control points. In addition, the blending functions are defined such that the curve passes through the first and last control points.

A B-spline curve has certain properties that make it suitable for defining an intuitive-looking path in cartesian space. First, it is variation diminishing, that is, it lies within the convex hull formed by the control points and does not oscillate wildly. Second, more control points can be added to change the shape of the curve without changing the degree of the curve. Third, control points only influence the shape of the curve locally, thus allowing us to define the shape of the curve in greater detail in special regions, such as corners and sharp turns, without changing the entire curve.

To ensure velocity continuity, we require that the curvature be continuous. For acceleration continuity, we require that the derivative of the curvature be continuous [15]. This in turn requires third order parametric continuity with respect to the parameter u. Hence, we choose quintic B-splines for specifying paths.

IV. Pass-Through-Door Task

The location of the wheelchair is specified by a *pose*, that is, a position and an orientation. The *pass-through-door* task is defined qualitatively as follows: start from rest on one side of the door, go through the door and reach a specified pose on the other side of the door with graceful motion (as defined in the introduction). It is assumed that the wheelchair passes through the goal on its way to some other target, hence it is in motion at the goal pose, that is, it does not stop at the goal pose. We assume that the world is static, that is, there are no moving obstacles. Thus, the only objects that the wheelchair needs to avoid are the walls and the door edges. We also assume that the robot has a local map available to it and knows where it is relative to the door, even when it is not facing the door.

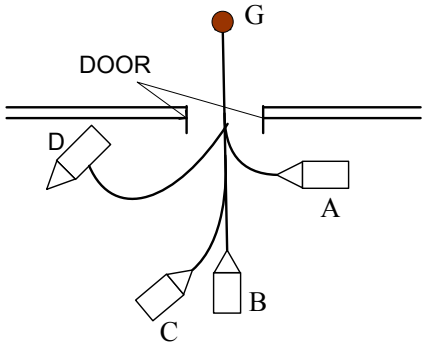


Fig. 2. Pass-Through-Door task

V. Implementation Details

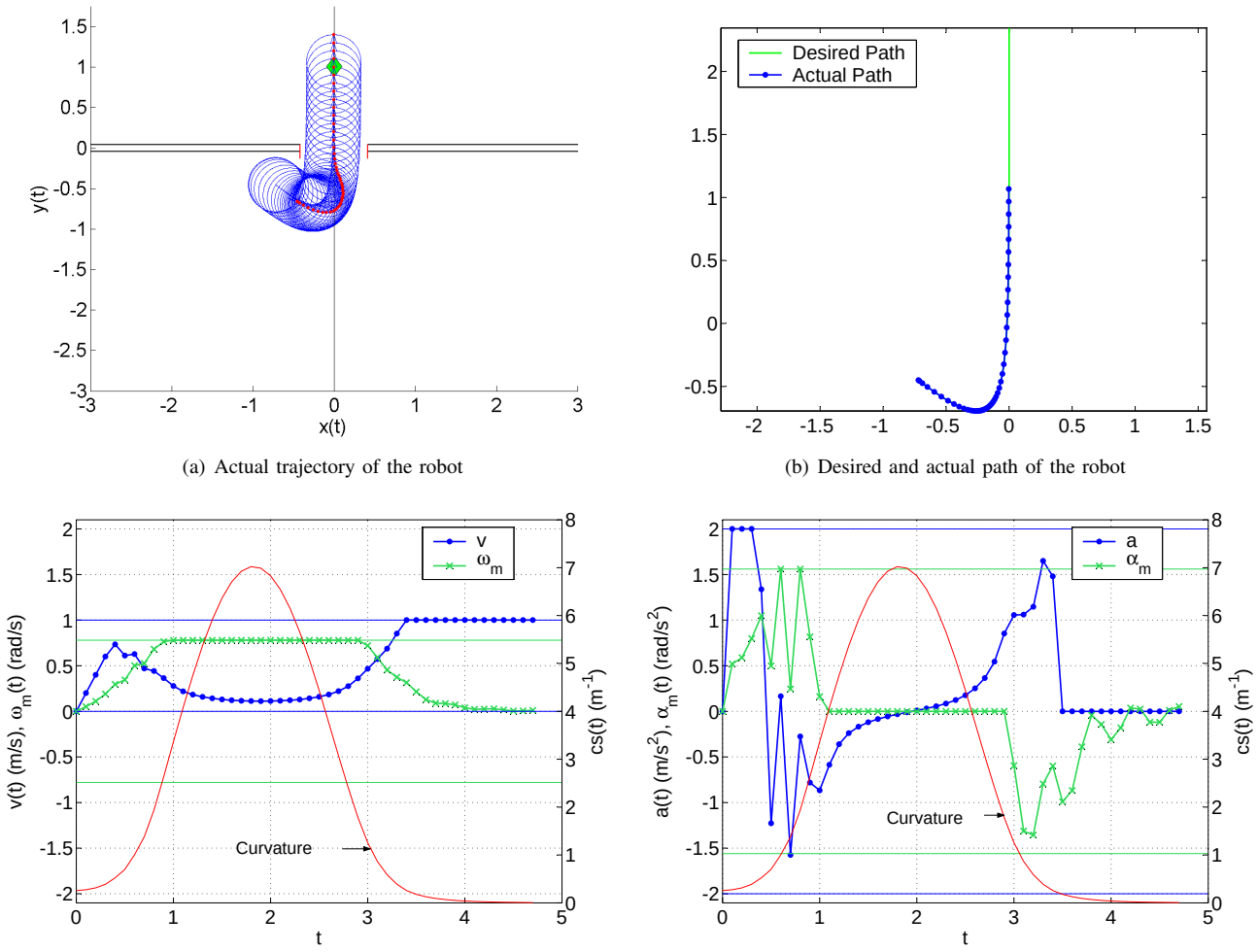
In this section, we discuss implementation results on a simulated system. We model the robot by Equation 1 along with the equations $\dot{v} = a$ and $\dot{\omega} = \alpha$ where a and α are the linear and angular accelerations respectively. We assume that the accelerations are constant for each time step. The robot model receives velocity and acceleration commands at the beginning of each time step and outputs the robot pose at the end of that time step. The controller takes the current position of the robot as input and generates velocity commands. The cycle time is 10 Hz, that is, the motion commands are generated every 0.1 seconds and sent to the robot model. Physically, we assume that the robot is circular in shape.

A. Parameter Values

The radius of the wheelchair is $r = 0.335$ m and the width of the doorway is $2.5r = 0.8375$ m. Thus, the maximum clearance between the robot and the door when the robot is at the center of the door is 8.37 cm. The constants in the control laws are chosen as $k_1 = 0.001$, $k_\delta = 5.0$ and $\gamma = 1.0$. It was determined experimentally that an approach angle $\theta_a = \pi$ yields good convergence to the path. The bounds on the velocities are set as $v \in (0, 1.0)$ m/s, $\omega_m \in (-0.78, 0.78)$ rad/s, $\dot{v} \in (-2.0, 2.0)$ m/s^2 and $\dot{\omega}_m \in (-1.56, 1.56)$ rad/s^2.

B. Control points of the B-spline for the pass-through-doorway task

The world frame is centered at the door with the positive X axis pointing to the right and the positive Y axis pointing toward the goal. Let (x_i, y_i) be the initial position of the wheelchair. The wheelchair must reach a goal on the other side of the door approximately on the mid line of the doorway. Hence, define the final position as $(0, 3r)$. We define the control points as $\langle (x_i, y_i), (0, -3r), (0, -2r), (0, 0), (0, 1.5r), (0, 7r) \rangle$. These



(a) Actual trajectory of the robot

(b) Desired and actual path of the robot

(c) Linear and angular velocities and curvature

(d) Linear and angular accelerations and curvature

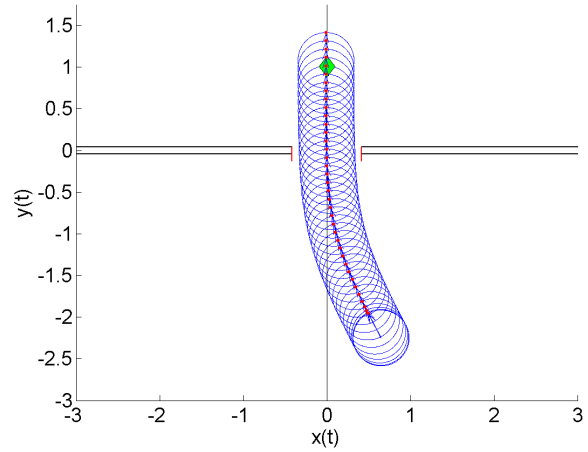
Fig. 3. Motion of the robot as it executes the pass-through-doorway task. The robot starts on one side of the door, very near the wall and makes a sharp turn to go out of the door. Each circular trace in Figure 3(a) shows the pose of the robot at a time step. The time steps are evenly spaced, so the robot is moving slower in places where the traces are closer together. Notice that the traces are closer together at the sharp turn and farther apart while going through the door. This shows that the robot passes through the doorway without slowing down.

control points define a path that is almost a straight line through the middle of the door, thus allowing the robot to pass straight through the center of the doorway. The curve defined by these points also enables a robot close to a wall near the door to first drive away from the wall toward the mid-line of the doorway and then go out the door as shown in Figure 3(a).

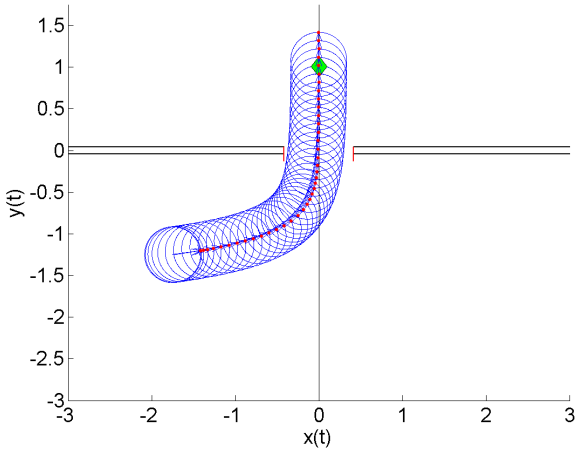
VI. Experiments and Evaluation

We tested our system for various representative starting poses of the robot and evaluated the behavior against the performance criteria formulated in the introduction. Figure 3 shows a case where the wheelchair is next to the wall very near the door and faces away from the door. This case is tricky because it involves moving along a high curvature ($7\ m^{-1}$) curve to go through the door. Figure 3(a) shows the cartesian trajectory of the robot and Figure 3(b) show the planned and actual paths. The velocities are shown in Figure 3(c). The path chosen requires the robot to constantly turn anti-clockwise to align itself with the path, hence the angular velocity is positive. Notice how the linear velocity drops at higher curvature values to enable the wheelchair to make the turn and is maximum on other parts of the curve. Notice also that the linear velocity starts to drop smoothly before arriving at the maximum curvature point. Analogously, in Figure 3(d), there is a large negative linear acceleration to slow down the wheelchair before the high curvature region and then a positive acceleration to speed it up after the turn.

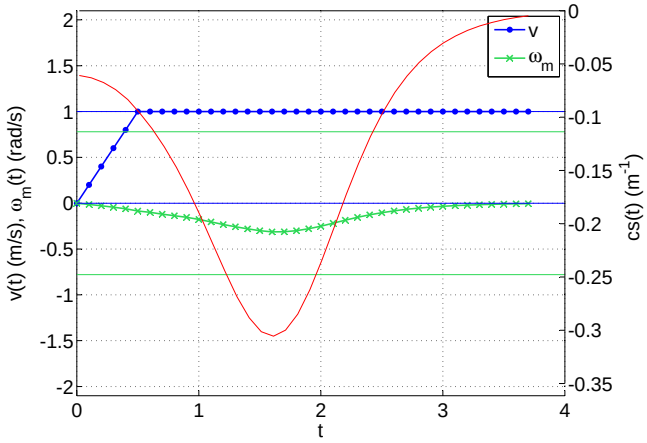
Two other trajectories of the robot and their corresponding velocity and acceleration curves are shown in Figure 4. The trajectory in Figure 4(a) has a very low curvature and once the robot has accelerated, it continues to move forward at its maximum linear velocity. In Figure 4(b), the robot moves forward with the maximum linear velocity possible. In the higher curvature part of the curve, the angular velocity rises to its maximum value so that the robot can continue moving as fast as possible while staying on the curve. Notice that in



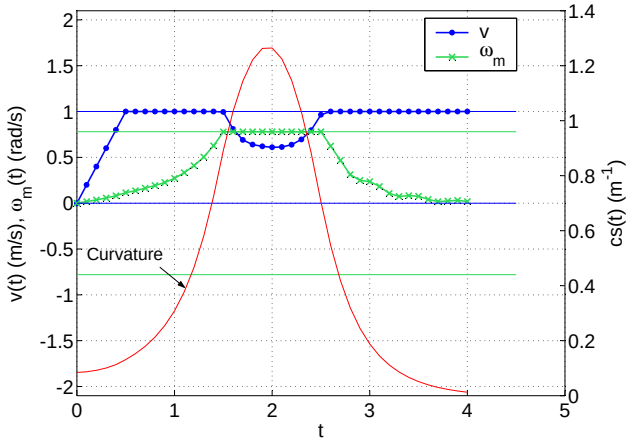
(a) A representative trajectory of the robot



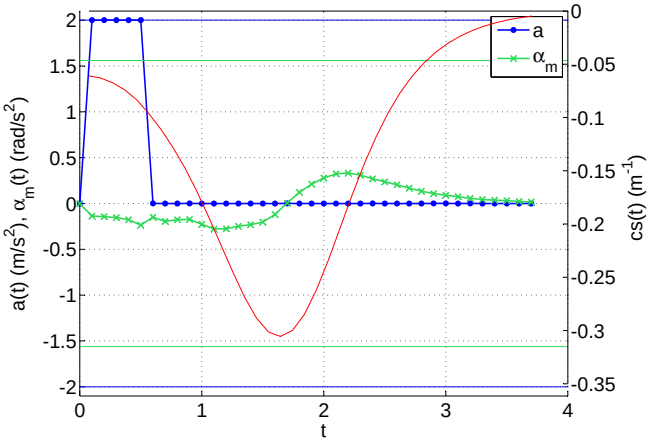
(b) Another representative trajectory of the robot



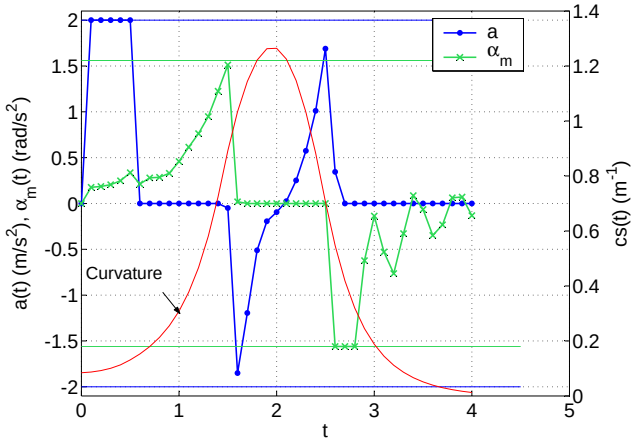
(c) Linear and angular velocities for the trajectory of Figure 4(a)



(d) Linear and angular velocities for the trajectory of Figure 4(b)



(e) Linear and angular accelerations for the trajectory of Figure 4(a)



(f) Linear and angular velocities for the trajectory of Figure 4(b)

Fig. 4. Motion of the robot along two representative paths as it executes the pass-through-doorway task.

all the cases shown above the robot's linear velocity increases to its maximum value as the curvature of the path decreases after the high curvature regions.

A. *Evaluation Results*

To evaluate for safety, we define minimum distance from the door and minimum distance from walls. If the minimum distance is greater than zero, the motion is collision-free.

TABLE I
EVALUATION RESULTS

Fig	Min Door Dist (cm)	Max Vels (m/s, rad/s)	Min Vels (m/s, rad/s)	Max Accs (m/s^2, rad/s^2)	Min Accs (m/s^2, rad/s^2)
3(a)	6.48	(1, 0.78)	(0, 0)	(2, 1.56)	(-1.58, -1.36)
4(a)	8.09	(1, 0.01)	(0, -0.31)	(2, 0.35)	(0, -0.64)
4(b)	4.66	(1, 0.78)	(0, 0)	(2, 1.51)	(-1.85, -1.56)

We have not implemented the bounds on jerk yet, hence we do not evaluate for those. For comfort, we check to see if the velocity and acceleration are within bounds. We define a minimum clearance of 2 cm from the door and walls as comfortable. To check for presence of oscillations, we do a simple visual evaluation since the number of samples is small. For a larger number of samples, a signal analysis to determine characteristic frequencies can be done.

The evaluation results for the three cases discussed above are shown in Table I. The results show that there are no collisions (safe), the velocity and acceleration bounds are not exceeded and there is sufficient clearance from the walls and the door (comfortable). The velocities change smoothly while the accelerations do not change very smoothly in the case of Figure 3(a). This means that there are large jerks in the motion: this is to be expected since we did not impose any jerk bounds in our formulation. Figures 3 and 4 show that the wheelchair does not slow down while passing through the door: instead it moves with its maximum possible velocity (fast).

VII. DISCUSSION

One issue worth considering is the importance of using the control law of Equation 8. The curves in Figures 3 and 4 show that the velocities obey the relation $\omega_m = vc_s$. One might ask whether it is sufficient to use the above control law rather than the one in Equation 8. The answer is that the relation above works well if the system is continuous time, that is, it is possible to instantaneously change the velocity of the robot so that it always stays on the curve. However, this is not the case for a real robot: the velocities can be sent to the robot only at discrete intervals, and hence the robot does not stay on the path at all instants. The law of Equation 8 drives the robot back to the path, whereas the simple control law $\omega_m = vc_s$ does not. This is illustrated by reducing the update rate from once per 0.1 seconds to once per 0.3 seconds. The law of Equation 8 lets the robot follow the path very closely while $\omega_m = vc_s$ does not.

Another issue is that while the solution converged for a large number of initial poses, it failed to converge for some time steps for some initial poses. One possible reason for non-convergence is that the last constraint in Equations 9 is approximate and should be replaced by the exact constraint derived from Equation 8.

VIII. CONCLUSIONS AND FUTURE WORK

This paper introduced the notion of graceful motion for a robotic wheelchair - motion that is safe, smooth, fast, and intuitive. We formulated some quantitative measures to characterize graceful motion, presented an approach for graceful navigation and implemented the approach on a specific task. Our results showed that the wheelchair satisfied the criteria for graceful motion. Future work consists of developing robust methods to learn to move gracefully that generalize to various tasks and environments and demonstrate results on the wheelchair.

IX. ACKNOWLEDGEMENT

This work has taken place in the Intelligent Robotics Lab at the Artificial Intelligence Laboratory, The University of Texas at Austin. Research of the Intelligent Robotics lab is supported in part by grants from the National Science Foundation (IIS-0413257, IIS-0713150, and IIS-0750011) and from the National Institutes of Health (EY016089).

REFERENCES

[1] A. Murarka, J. Modayil, and B. Kuipers, "Building local safety maps for a wheelchair robot using vision and lasers," in *Canadian Conference on Computer and Robot Vision*, 2006, p. 25.

[2] P. Beeson, M. MacMahon, J. Modayil, A. Murarka, B. Kuipers, and B. Stankiewicz, "Integrating multiple representations of spatial knowledge for mapping, navigation, and communication," in *Interaction Challenges for Intelligent Assistants*, 2007, pp. 1–9.

[3] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics and Automation Magazine*, vol. 4, no. 1, 1997.

[4] R. Simmons, "The curvature-velocity method for local obstacle avoidance," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 4, 1996, pp. 3375–3382.

[5] J. Minguez, J. Osuna, and L. Montano, "A divide and conquer strategy based on situations to achieve reactive collision avoidance in troublesome scenarios," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2004, pp. 3855–3862.

[6] J. Latombe, *Robot Motion Planning*. Kluwer Academic Publishers, Boston, MA, 1991.

[7] E. Rimon and D.E.Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Transactions on Robotics and Automation*, no. 5, pp. 501–518, 1992.

[8] K. Konolige, "A gradient method for realtime robot control," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2000, pp. 639–646.

[9] T. Fraichard and A. Scheuer, "From reeds and shepp's to continuous-curvature paths," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 6, pp. 1025–1035, 2004.

[10] A. Micaelli and C. Samson, "Trajectory tracking for unicycle-type and two-steering-wheels mobile robots," INRIA, Sophia-Antipolis, Tech. Rep. 2097, 1993.

[11] L. Lapierre, D. Soetanto, and A. Pascoal, "Nonsingular path following control of a unicycle in the presence of parametric modelling uncertainties," *International Journal of Robust Nonlinear Control*, pp. 485–503, 2006.

[12] A. Piazzi and C. G. L. Bianco, "Quintic G^2-splines for trajectory planning of autonomous vehicles," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, 2004, pp. 620–625.

[13] E. Magid, D. Keren, E. Rivlin, and I. Yavneh, "Spline-based robot navigation," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 2296–2301.

[14] M. Mortensen, *Geometric Modeling*. Wiley & Sons, New York, 1985.

[15] B. Barsky and T. DeRose, "Geometric continuity of parametric curves: three equivalent characterizations," in *Proceedings of the IEEE Conference on Computer Graphics and Applications*, vol. 9, 1989, pp. 60–69.