# ATRS - A Technology-Based Solution to Automobility for Wheelchair Users

Chao Gao[1], Ira Hoffman[2], Thomas Panzarella[2], and John Spletzer[1]

[1] Lehigh University, Bethlehem, PA, USA
   spletzer@cse.lehigh.edu
[2] Freedom Sciences, LLC, Philadelphia, PA, USA
   tpanzarella@freedomsciences.com

**Summary.** In this paper, we present the Automated Transport and Retrieval System (ATRS). ATRS represents an alternative to van conversions for automobile drivers with lower body disabilities. It employs robotics and automation technologies that integrate into a standard mini-van or sport utility vehicle (SUV). At the core of ATRS is a "smart" wheelchair that navigates between the driver's position and a powered lift at the rear of the vehicle - eliminating the need for an attendant. From an automation perspective, autonomously docking the wheelchair onto the lift platform presented the most significant technical challenge during system development. This was driven by geometry constraints, which limited clearance between the chair wheels and the lift platform rails. To solve this problem, we employed an LMS291 LIDAR in conjunction with an Extended Kalman Filter for reliable and accurate wheelchair localization. Coupled with a hybrid controller design, the system has proven to be exceptionally robust. This was validated through extensive simulation and experimental results, culminating in a three-day demonstration at the 2006 World Congress and Exposition on Disabilities where the system completed over 300 consecutive cycles without a failure.

## 1 Introduction and Motivation

According to the U.S. Bureau of Transportation, over six million people with disabilities have difficulties in obtaining the transportation they need [1]. This is a major contributor to the unemployment rate of the disabled population nationally, estimated at over 65% by the U.S. Census Bureau [2].

A van conversion currently represents the sole personal transportation solution for an individual in a wheelchair. Van conversions start with a standard van produced by a major automotive manufacturer. The van is subsequently modified or "converted" by another company, usually a specialized mobility equipment manufacturing company or mobility dealer. The modifications are permanent, and include extensive changes to the chassis, frame, and interior. Typical modifications include removing and lowering the vehicle floor, and relocating/replacing major subsystems such as the gas tank, fuel system, and heating/cooling systems of the vehicle [3]. While enabling independent mobility, van conversions represent a costly and unsafe transportation solution for wheelchair users.

To eliminate these shortcomings, we have developed a technology-based alternative to van conversions for wheelchair users: the Automated Transport and Retrieval

System (ATRS). ATRS employs robotics and automation technologies, and can be integrated into a standard mini-van or sport utility vehicle (SUV). At the core of ATRS is a "smart" wheelchair system that autonomously navigates between the driver's position and a powered lift at the rear of the vehicle. A primary benefit of this paradigm is the operator and chair are separated during vehicle operations as well as entry/exit. This eliminates the potential for injuries or deaths caused by both improper securement (as the operator is seated in a crash tested seat system) as well as lift malfunctions. Furthermore, by eliminating the *drastic* and *permanent* vehicle modifications associated with van conversions, ATRS will cost significantly less.

## 2    Related Work

Extensive work has been done in order to increase the safety levels of power wheelchairs while minimizing the level of human intervention. In these systems, the human operator is responsible for high-level decisions while the low-level control of the wheelchair is autonomous.

The Tin Man system [4], developed by the KISS Institute, automates some of the navigation and steering operations for indoor environments. The Wheelesley project [5], based on a Tin Man wheelchair, is designed for both indoor and outdoor environments. The chair is controlled through a graphical user interface that has successfully been integrated with an eye tracking system and with single switch scanning as input methods. The TAO Project [6] provides basic collision avoidance, navigation in an indoor corridor, and passage through narrow doorways. The system also provides landmark-based navigation that requires a topological map of the environment. The NavChair assistive wheelchair navigation system [7] uses feedback from ultrasonic sensors and offers obstacle avoidance, door passage, and wall following modes. More recently, the SmartChair [8] uses a virtual interface displayed by an on-board projection system to implement a shared control framework that enables the user to interact with the wheelchair while it is performing an autonomous task.

A common theme in the above research is the robotics technology has been applied to assist or augment the skills of the chair operator. In contrast, the ATRS wheelchair is in fact capable of autonomous vehicle navigation in outdoor environments. This can be realized because the operator is never seated in the chair during autonomous operations, and the chair always operates in the vicinity of the automobile. The former constraint mitigates operator safety issues, while the latter provides significant, invariant landmarks/features in an otherwise unstructured environment.

## 3    System Overview

In describing the ATRS operational procedures, we refer to Figure 1. When the operator returns to his/her automobile, a keyless entry is used to both unlock the vehicle and to deploy the traversing driver's seat. The operator then positions the wheelchair, and performs a seat-to-seat transfer (pose A). After this, the wheelchair is deployed to the rear of the vehicle (pose B). In our proof-of-concept system, this side traversal was
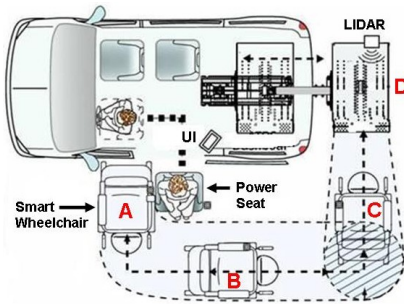
**Fig. 1.** ATRS concept diagram illustrating the primary system components

completely autonomous [9]. In the current system – referred to colloquially as "ATRS-Lite" – the wheelchair is remotely controlled by the vehicle operator via a joystick located at the UI. Once the chair enters the LIDAR's field-of-view at the rear of the vehicle (pose C), it is automatically tracked. The UI then cues the operator to place the wheelchair into "docking" mode. This enables the van-side computer to transmit real-time control inputs to the chair over a dedicated RF link for reliable docking (locking in place) onto the lift platform (pose D). With the chair docked, the operator actuates the lift via the UI – stowing the platform and chair into the vehicle cargo area. The process is repeated in reverse when disembarking from the automobile. We should emphasize that when not operating autonomously, the ATRS wheelchair is placed in "manual mode," and operates no differently than any other powered wheelchair

The primary focus of this paper is the development of a reliable, autonomous means for docking (and undocking) the ATRS wheelchair onto (and off of) the vehicle's lift platform. Our current approach employs a SICK LMS291-S14 LIDAR system in conjunction with an Extended Kalman Filter (EKF) for estimating the chair pose. EKF techniques for feature-based mobile robot localization were pioneered by Durrant-Whyte and others (*e.g.*, [10]), and such an approach has also proven to be well suited for our application.

## 4   Wheelchair Localization

To reliably execute docking under a broad range of environmental conditions, wheelchair localization requires both robust feature segmentation as well as accurate pose estimation with respect to the lift platform. The platform mounted LMS291 LIDAR system provides bearing, range, and reflectivity measurements that are leveraged for robust feature segmentation. The positions of these features - in conjunction with the control inputs to the chair - are then used as input to an Extended Kalman Filter that estimates the wheelchair pose over time. Details of the localization process are as follows.

### 4.1   Feature Segmentation

The primary sensor used for estimating the wheelchair pose with respect to the lift platform is a SICK LMS291 LIDAR. Figure 2 (left) illustrates a typical integration of
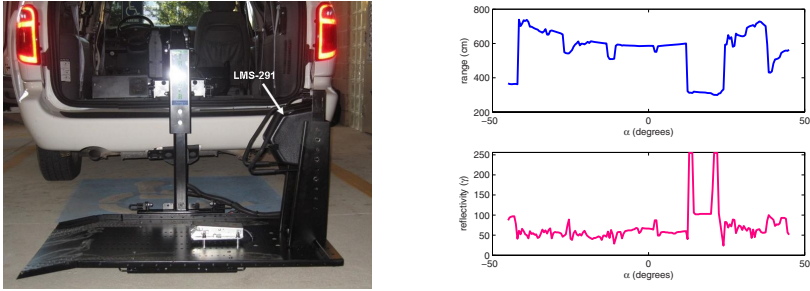
**Fig. 2.** (left) The $Tracker^{TM}$ lift platform with integrated LMS291. The LIDAR housing provides both environmental protection, as well as improves the system SNR. (right) Range (top) and reflectivity (bottom) data from a single LIDAR scan. As was typical, reflectivity measurements alone were adequate for segmenting the two target features.

the LIDAR into the vehicle lift platform. The LMS291 measures the line-of-sight range to objects in the environment over a 90° degree field of view with a discretization of 0.5°. Each of these measurements can be written as a tuple of the form $\mathbf{z}_m = [r, \alpha, \gamma]_m^T$, $m = 0...180$, where $r_m$ and $\gamma_m$ denote the measured range to and reflectivity of the $m^{th}$ feature at a bearing of $\alpha_m = \frac{m}{2} - 45°$ with respect to the LIDAR sensor frame $\mathcal{L}$. We exploit the reflectivity measurements to greatly simplify the segmentation problem. A pair of target features $\{t_l, t_r\}$ made from retro-reflective material are permanently affixed to the wheelchair chassis. When imaged by the LIDAR, a significant portion of the incident beam is reflected directly back to the detector, saturating the photo-diode (Figure 2 (left)). This allows a simple threshold on reflectivity $\gamma_{min}$ to be used as the primary filter for segmenting the target features. An additional level of filtering is based upon a range constraint $r_{max}$. As the wheelchair is presented in the immediate vicinity of the lift platform, targets at excessive ranges (*e.g.* > 4 meters away) can immediately be disqualified from potential features of interest. From these two filters and assuming a ground plane constraint, we construct a valid feature set

$$F = \begin{bmatrix} r_n \cos \alpha_n \\ r_n \sin \alpha_n \end{bmatrix}, \ s.t. \ r_n < r_{max}, \ \gamma_n > \gamma_{min} \qquad (1)$$

A final level of filtering exploits a priori knowledge of the relative geometry of the wheelchair targets. Candidate targets $T \supseteq \{t_l, t_r\}$ are identified via clustering the candidate feature set $F$ in Euclidean space using the actual target size (plus a tolerance) as a constraint. Pairs of targets $t_i, t_j \in T$ are then examined using the actual target distance $||t_l - t_r||$ as a binary filter to identify the correct target pair in the LIDAR scan. If a valid target pair cannot be identified, the operator is alerted to take corrective action (*i.e.*, reposition the chair) and the process repeated.

## 4.2   Pose Estimation

An Extended Kalman Filter (EKF) is used to estimate the chair pose. The prediction step employs feedback from the on-board encoders, while the correction step leverages

the target positions recovered from the segmentation process. The process model for the EKF is based upon the standard kinematic motion model for the differential-drive wheelchair, and is not covered here in the interest of brevity. Instead, we focus upon the measurement update phase of the EKF where position estimates of the two retro-reflective targets are used to correct the pose estimate. A straight-forward approach to modeling the LIDAR measurements would be to derive the measurement Jacobian $H$ based upon the range and bearing measurements. There are potential shortcomings with such an approach. The corresponding measurement equations (for bearings in particular) are highly non-linear, and are not well modeled by a first-order approximation. Ultimately, this could lead to poor filter performance. Instead, with each LIDAR scan we directly estimate the position of the features in the world frame. This is straightforward from the available LIDAR measurements

$$\mathbf{z}(t_k) = \begin{bmatrix} x_l \\ y_l \end{bmatrix}_{t_k} + r_i \begin{bmatrix} \cos(\alpha_i + \theta_l) \\ \sin(\alpha_i + \theta_l) \end{bmatrix}_{t_k} \qquad i \in \{1, 2\} \tag{2}$$

where $[z_{x_i}, z_{y_i}]^T$ corresponds to the position of the $i^{th}$ feature in our world frame $\mathcal{W}$, $[x_l, y_l, \theta_l]^T$ reflect the coordinate transformation from the LIDAR sensor frame $\mathcal{L}$ to $\mathcal{W}$, and $(r_i, \alpha_i)$ denote the range and bearing measurements to the $i^{th}$ target with respect to $\mathcal{L}$. These feature positions are then are used directly by the filter. This is akin to using GPS measurements in the filter rather than the raw range measurements from which they are formed - a common practice in mobile robotics. The corresponding measurement equations for the filter can then be written as

$$\mathbf{h}_i(t_k) = \begin{bmatrix} x \\ y \end{bmatrix}_{t_k} + \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}_{t_k} \begin{bmatrix} a_i \\ b_i \end{bmatrix} \tag{3}$$

where $[a_i, b_i]^T$ denotes the fixed position of the $i^{th}$ target in the robot frame $\mathcal{R}$, and $\theta$ corresponds to the *predicted* wheelchair orientation at the $k^{th}$ time-step. The resulting measurement Jacobian is

$$H_i(t_k) = \begin{bmatrix} 1 & 0 & -a_i \sin\theta - b_i \cos\theta \\ 0 & 1 & a_i \cos\theta - b_i \sin\theta \end{bmatrix} \tag{4}$$

One issue remaining is properly modeling the covariance of the resulting sensor measurements. Each range and bearing measurement defines a new coordinate frame with basis vectors $u_1 = [\cos\alpha_i, \sin\alpha_i]^T$, $u_2 = [\sin\alpha_i, -\cos\alpha_i]^T$. The uncertainty in the $u_1$ direction corresponds directly to the variance of the range measurement $\sigma_{u_1}^2 = \sigma_r^2$. Uncertainty in the $u_2$ direction is a function of the uncertainty of the bearing angle measurement where $\sigma_{u_2}^2 = r_i^2 \sin^2 \sigma_\alpha$. Noting that for the LMS291, $\sigma_\alpha$ is in fact quite small ($< 1°$), this can be very well approximated by $\sigma_{u_2}^2 = r_i^2 \sigma_\alpha^2$. Since this transformation is linear in terms of $\sigma_\alpha$, we expect the characteristics of the (assumed Gaussian) noise of the bearing measurements to be preserved.

The last step is to transform the corresponding covariance matrix to $\mathcal{W}$. This can readily be accomplished through the similarity transform

$$R_i(t_k) = T_i(t_k) \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & r_i^2 \sigma_\alpha^2 \end{bmatrix} T_i(t_k)^T \tag{5}$$

where

$$T_i(t_k) = \begin{bmatrix} \cos(\alpha_i + \theta_l) & -\sin(\alpha_i + \theta_l) \\ \sin(\alpha_i + \theta_l) & \cos(\alpha_i + \theta_l) \end{bmatrix}_{t_k} \tag{6}$$

and where $R_i$ is the measurement covariance for the $i^{th}$ target. The correction step for the EKF can then be written in standard form.

## 5    Wheelchair Control

From an automation perspective, two aspects to the control problem must be considered: *motor* control and *motion* control. The motion controller generates higher level velocity commands vanside based upon the current chair pose as estimated via the localization system presented in Section 4. These are in turn transmitted to the powerchair over a dedicated RF link, which regulates the wheel velocities to achieve the objective linear and angular velocities for docking.

### 5.1    Motor Control

The wheelchair employs a differential drive system where the motion controller transmits objective linear and angular velocities, which are in turn mapped to wheel velocities. These are then regulated via a PID controller implemented in software on the chair's embedded PC. Feedback to the PID is provided via high-resolution quadrature encoders that measure right and left wheel travel $(\Delta\phi_L, \Delta\phi_R)$ at 100Hz. To properly model motor controller performance, experiments were conducted to characterize the latency between controller input and chair actuation. These trials indicated typical latencies of 500-600ms between motor controller input and wheelchair actuation. When considering the nominal linear velocity of the chair was 40 cm/s, and the clearances associated with docking were on the order of 4 cm, such a delay was significant. This influenced the design of the motion controller, as discussed in the sequel.

### 5.2    Motion Control

Motion controller design was influenced by real-world constraints associated with system use. These included not only the controller latency, but also docking clearances and the constrained ground area adjacent to the vehicle for navigation. As such, our motion planner employed a hybrid control design consisting of two primary controller modes: course-correction, and path-following. In this paradigm, gross y-position errors were first corrected (when necessary) in the course-correction phase before proceeding to path-following for docking. We now describe each mode in greater detail.

**Path-Following Phase**

Path-following was the primary controller mode, and the one used for docking onto the lift platform. It employs a traditional PD controller derived using I/O feedback linearization techniques

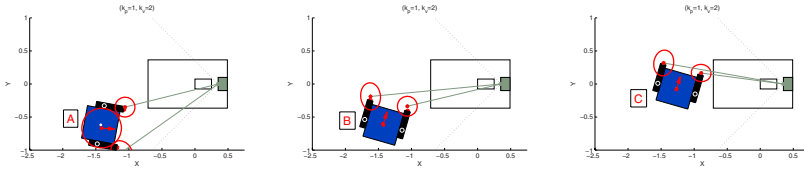$$\omega = -k_v \tan\theta - \frac{k_p y}{v \cos\theta} \tag{7}$$

**Fig. 3.** Docking trial simulation illustrating controller course-correction to eliminate gross y-position errors. At position C, the wheelchair switches to path-following mode for final docking. In this example, $\theta_{max} = 75°$.

where $\omega$ and $v$ are the desired linear and angular velocities transmitted to the chair, $v$ is assumed piecewise constant, and $k_v, k_p$ are positive controller gains. For safety considerations, we specify maximum linear and angular velocities $[v_{max}, \omega_{max}]^T$ for the chair. Typical values were 0.4 m/s and 0.8 rad/s in practice (significantly less than what can be achieved by the actual hardware). To accommodate these limits, we borrow from [11] and constrain the actual controller inputs to

$$\omega_{act} = \mathcal{S}(\omega) \arg\min\{|\omega|, \omega_{max}\} \;,\; v_{act} = \tfrac{\omega_{act}}{\omega} v_{max} \tag{8}$$

where $\mathcal{S}()$ corresponds to the *sign* function. These constraints ensure that while the wheelchair will no longer follow the same trajectory specified by (7), it will follow the same path while protecting against actuator saturation.

One further refinement was made immediately preceding path-follwing mode by an orientation correction. The intent was to find an initial orientation $\theta^*$ such that the magnitude of $\omega_0$ is minimized - and ideally zero. From (7), we obtain two possible solutions:

$$\theta^* = \left\{ -\arcsin\left(\frac{k_p y}{k_v v}\right), -\arcsin\left(\frac{k_v v}{k_p y}\right) \right\} \tag{9}$$

So, for the case where $|(k_p y)/(k_v v)| \leq 1$ there is an initial orientation for our path follower that requires zero initial angular velocity. Fortunately, our configuration parameters allow such an orientation to be readily achieved. Thus, all initial orientation error can be removed prior to initializing the path-follower controller.

**Course-Correction Phase**

To enhance ATRS docking reliability, a course-correction mode is also incorporated to address gross y-position errors. This controller phase is activated after initial localization in autonomous mode *only if* it is determined that the path-following mode would be at risk for failing to dock the chair at the handoff location provided by the operator (*e.g.*, for large y-position errors with the chair left too near the lift platform). In this event, we again exploit the chair's two degrees of mobility to align the chair along the x-axis in our world frame. This is accomplished through the following set of control inputs.

$$d\theta_R = -\theta_{max}\mathcal{S}(y_0) - \theta_0 \;,\; dx_R = \frac{abs(y_0)}{\sin\theta_{max}} \tag{10}$$

where $y_0, \theta_0$ denote the initial y-position and orientation of the robot, respectively, and $\theta_{max}$ corresponds to a maximum allowable orientation angle for the wheelchair that

ensures both retro-reflective targets will be visible to the LIDAR (somewhat less than 90°). The effect of these inputs is illustrated in simulation at Figure 3. The first reorients the wheelchair from its initial pose (A) to one more normal to the x-axis (B). After this, the chair translates to approximately $y = 0$ (C). At this point, the controller switches to path-following, where the initial orientation correction will reorient the wheelchair for docking. The net result is a dramatic reduction in the settling distance, which also reduces the impact of latency on controller robustness.
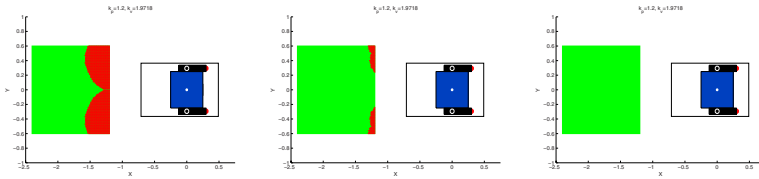


**Fig. 4.** Controller performance with the PD control law (left), orientation correction integrated (center), and course-correction added (right). The latter eliminated residual poses associated with docking failure. Simulation resolution was 1 $cm^2$.

## 6   Simulation Results

Docking performance was first extensively evaluated in simulation. Our simulation model echoed the real-world system characteristics to the extent possible. It integrated the EKF for localization, and included (incompletely) modeled estimates for controller latency, process noise in the odometry system, and measurement noise for the LIDAR. Monte-Carlo simulations were then run over the range of feasible poses to identify potential failure conditions for docking. These were also useful in demonstrating the evolution of the controller strategy.

Figure 4 illustrates the results of a Monte-Carlo simulation used to assess controller performance. Green and red dots reflected whether the corresponding initial position resulted in a successful or failed dock attempt, respectively. For purposes of this simulation, success corresponded to orientation and y-position errors less than 10° and 5 cm, respectively, before the chair reached the ramp of the lift platform. Figure 4 (left) reflects the performance of the path-follower component from (7) alone. Controller performance is improved when path-following is preceded by an initial orientation correction (center), while all failure modes were eliminated with the integration of the course-correction component (right).

As a testament to the fidelity of the simulation, the gains used for the path-follower controller on the actual vehicle were nearly identical to those obtained through the simulation process.

## 7   Experimental Results and Future Work

Over the past six months, the beta ATRS has been tested across a range of conditions. This included three days of continuous demonstrations at the World Congress Exposition on Disabilities (WCD 2006) in November 2006. Conference participants were also

**Fig. 5.** ATRS demonstration at WCD2006. The operator performs a seat-to-seat transfer (left), and then remotely controls the wheelchair to the vicinity of the lift platform (center). At this point, the LIDAR tracks the chair while sending real-time control inputs over a dedicated RF link. Autonomous operations conclude with the chair successfully docked (right).

given the opportunity to test the system. Over 300 cycles of docking and undocking were conducted during this time without a single failure. A sample trial is illustrated at Figure 5. A video of this same trial can be viewed at $http://vader.cse.lehigh.edu/$

While we are optimistic that this framework is sufficiently robust under real-world conditions, additional work remains. In the shorter term, this includes integrating a gyroscope with the odometry system to detect wheel slippage, as well as actuating the LIDAR pitch to relax our ground-plane assumption and reduce the fiducial size. In the longer term, we are also investigating an active vision system to be used in conjunction with (or as an alternative to) the LMS291.

## Acknowledgments

## References

1. National Council on Disabilities. In: The Current State of Transportation for People with Disabilities in the United States, 1331 F Street, NW, Suite 850, Washington, D.C. 20004 (June 2005)
2. Stern, S., Brault, M.: Disability Data From the American Community Survey: A Brief Examination of the Effects of Question Redesign in 2003 (January 2005)
3. Wood, T.: When your plan is a van: Decisions, Decisions A Guide to Asking the Right Questions and Choosing the Best Vehicle for You. Quest - Journal of the Muscular Dystrophy Association 11(2) (March-April 2004)
4. Miller, D., Slack, M.: Design and testing of a low-cost robotic wheelchair prototype. Autonomous Robots 1(3) (1995)

5. Yanco, H.: Wheelesley, a robotic wheelchair system: indoor navigation and user interface. In: Assistive Technology and Artificial Intelligence. LNCS (LNAI), pp. 256–268 (1998)
6. Gomi, T., Griffith, A.: Developing intelligent wheelchairs for the handicapped. In: Assitive Technology and Artificial Intelligence. LNCS (LNAI), pp. 150–178 (1998)
7. Simpson, R., Levine, S.: Automatic adaptation in the NavChair assistive wheelchair navigation system. IEEE Transactions on Rehabilitation Engineering 7(4), 452–463 (1999)
8. Parikh, S.P., Grassi, V., Kumar, V., Okamoto, J.: Incorporating user inputs in motion planning for a smart wheelchair. In: Proc. of the IEEE International Conference on Robotics and Automation (ICRA), New Orleans, LA, pp. 2043–2048 (2004)
9. Sermeno-Villalta, H., Spletzer, J.: Vision-based Control of a Smart Wheelchair for the Automated Transport and Retrieval System. In: Proceedings of the 2006 IEEE International Conference on Robotics and Automation (ICRA 2006), Orlando, FL, USA (May 2006)
10. Leonard, J., Durrant-Whyte, H.F.: Mobile robot localization by tracking geometric beacons. IEEE Trans. on Robotics and Automation 7(3), 376–382 (1991)
11. Oriolo, G., Luca, A., Vendittelli, M.: Wmr control via dynamic feedback linearization: Design (2002)