A Segway RMP-based robotic transport system

Hoa G. Nguyen,[a*] Greg Kogut,[a] Ripan Barua,[a] Aaron Burmeister,[a]
Narek Pezeshkian,[a] Darren Powell,[a] Nathan Farrington,[a]
Matt Wimmer,[b] Brett Cicchetto,[b] Chana Heng,[b] and Velia Ramirez[b]

[a] Space and Naval Warfare Systems Center, San Diego, CA 92152-7383
[b] High Tech High School, San Diego, CA 92106-6025

ABSTRACT

In the area of logistics, there currently is a capability gap between the one-ton Army robotic Multifunction Utility/Logistics and Equipment (MULE) vehicle and a soldier's backpack. The Unmanned Systems Branch at Space and Naval Warfare Systems Center (SPAWAR Systems Center, or SSC), San Diego, with the assistance of a group of interns from nearby High Tech High School, has demonstrated enabling technologies for a solution that fills this gap. A small robotic transport system has been developed based on the Segway Robotic Mobility Platform^{TM} (RMP).

We have demonstrated teleoperated control of this robotic transport system, and conducted two demonstrations of autonomous behaviors. Both demonstrations involved a robotic transporter following a human leader. In the first demonstration, the transporter used a vision system running a *continuously adaptive mean-shift filter* to track and follow a human. In the second demonstration, the separation between leader and follower was significantly increased using Global Positioning System (GPS) information. The track of the human leader, with a GPS unit in his backpack, was sent wirelessly to the transporter, also equipped with a GPS unit. The robotic transporter traced the path of the human leader by following these GPS breadcrumbs.

We have additionally demonstrated a robotic medical patient transport capability by using the Segway RMP to power a mock-up of the Life Support for Trauma and Transport (LSTAT) patient care platform, on a standard NATO litter carrier. This paper describes the development of our demonstration robotic transport system and the various experiments conducted.

Keywords: Segway Robotic Mobility Platform, RMP, LSTAT, logistics, visual following, GPS breadcrumb

1. INTRODUCTION

The Segway RMP is a two-wheel robotic mobility platform based on the self-balancing Segway Human Transporter (HT)—see Fig. 1. Developed to free DARPA Mobile Autonomous Robot Software (MARS) researchers from many limitations of current platforms, the RMP can travel at high speeds (up to 13 km/h), is capable of indoor and outdoor operation, has zero turning radius, can provide a human-height sensor point of view and manipulation, and has a payload capacity of over 100kg. The RMP can travel on the order of 15 km between recharges from any standard AC outlet. The hardware interface to the RMP is via CAN bus (with a USB interface to be available soon); the software interface is via a simple high level application programming interface. This eliminates the need for the usual software cross compiling and downloading to an embedded processor.

The RMP's dynamic self-balancing feature and its human-like dimensions invite a number of unique applications, from acting as a simpler alternative to robotic legs for humanoid robots, to competing with and alongside humans in soccer matches. These and other current applications, as well as the history behind the development of the RMP, are described in a companion paper at this conference.[1] Here we concentrate on three military applications for the RMP explored at SSC San Diego that do not use the platform's self-balancing ability. All three applications capitalized on the RMP's ruggedness, size, and payload (weight pulling) capacity.

[*] E-mail: hoa.nguyen@navy.mil

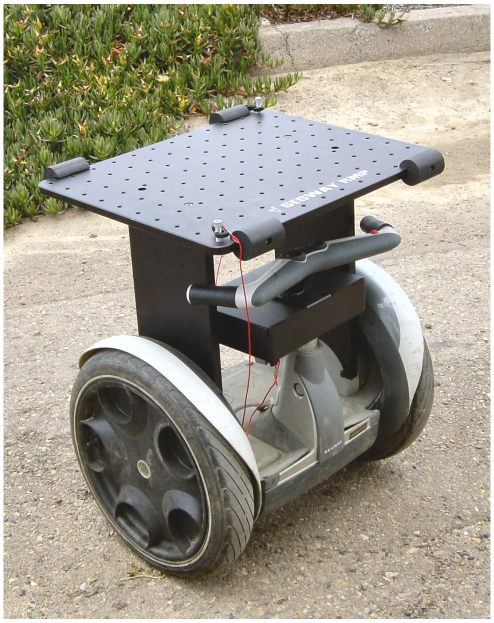


Figure 1. The Segway RMP.

2. ROBOTICIZED LSTAT

The Life Support for Trauma and Transport (LSTAT, see Fig. 2) is a portable intensive care system developed by Integrated Medical Systems, Inc.[2] The system allows for an injured person to be treated in situations and/or at locations where advanced medical equipment would not otherwise be available. The system includes a ventilator, a suction system, an oxygen system, an infusion pump, a physiological monitor, a clinical blood analyzer, and a defibrillator.

The portability and life saving capabilities of the LSTAT make it a valuable asset in military conflicts as well as natural disasters for the treatment and transportation of injured persons. The weight of the LSTAT and litter is approximately 80kg; add a patient weighing another 80kg and a total of 160kg must be transported. Currently, two to four persons are required to transport an LSTAT. To facilitate the transport and care of large numbers of injured persons, we explored the use of the Segway RMP as a robotic transport system for the LSTAT. The robotic transport system saves care givers valuable time, allowing them to focus on providing immediate medical treatment while the robot provides the more logistical task of transportation.

We constructed the roboticized LSTAT prototype, shown in Fig. 3, from a Segway RMP, a hitch, a standard NATO litter carrier, and an LSTAT shell fully weighed with sand bags. (An actual LSTAT was not used due to the cost involved.) The system successfully demonstrated patient transport through interior corridors, on sidewalks, and over paved streets. The system also performed surprisingly well in an all-terrain environment. However, the high center of gravity created a stability problem when driving over extremely rough terrain with loose soil and dried scrub brush.

Lessons we have learned after extensive testing of our prototype include:

- An improved prototype would not use the NATO Litter Carrier as the supporting portion of the system. The carrier had no suspension and was susceptible to tipping due to its high center of gravity.

- Currently the hitch is about 50 cm above the drive axis of the RMP. This hitch location induced a large moment on the hitch bearings, causing a crack in the housing after one test over particularly rough terrain. An improved prototype would bring the hitch down as close to the drive axis as possible to reduce moment forces.
- Moving the system's center of gravity forward—towards the RMP—would give the drive wheels more traction.

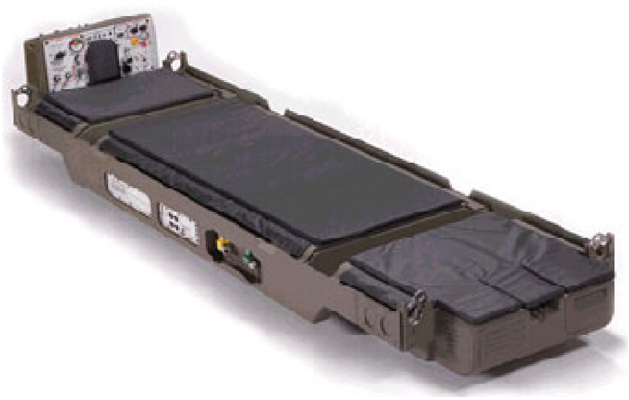


Figure 2. The LSTAT.[2]

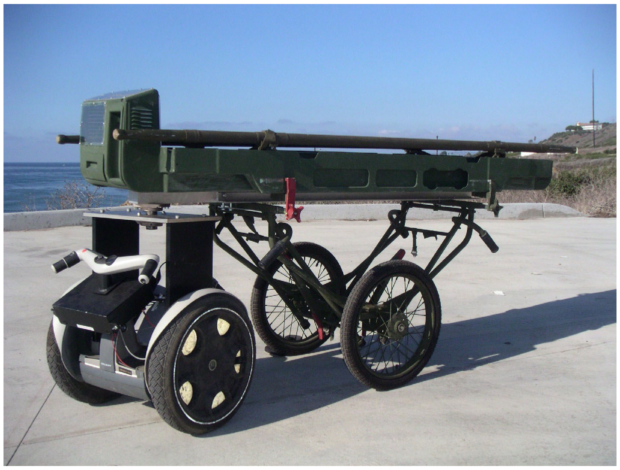


Figure 3. SSC San Diego's roboticized LSTAT prototype.

We had planned on adding autonomous behaviors to the roboticized LSTAT. However, during the last teleoperated outdoor test, the combination of high center-of-gravity, rough terrain, loose soil, and dried brush caused the platform to tip and fall on its side. With the search for a lower center-of-gravity LSTAT carrier, we realized that such a carrier could perform many other functions, and the effort segued into technology demonstrations for a general-purpose logistics transport system (Fig. 4). The system consists of the Segway RMP and a general-purpose trailer cart, mounted using a hitch placed close to the RMP's wheel axis. Two autonomous behaviors were subsequently explored on this platform: vision-based following and GPS breadcrumb following. Both can be equally applied to a roboticized LSTAT.

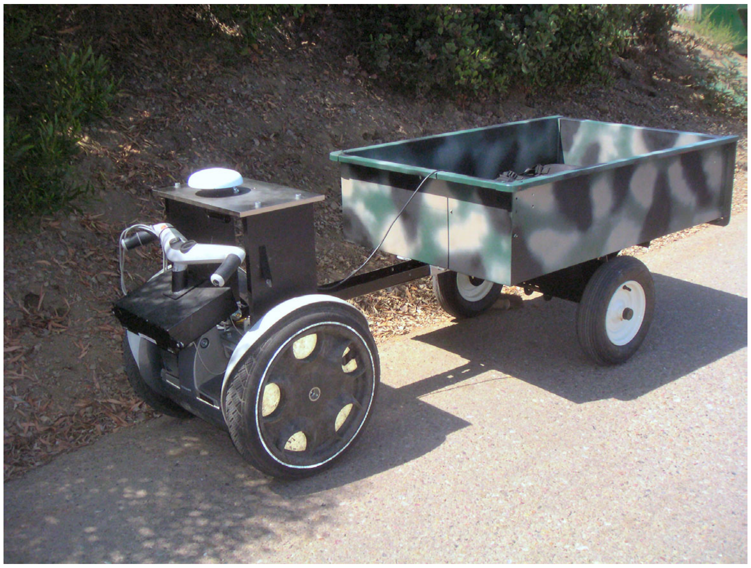


Figure 4. The general-purpose logistics transport system.

3. VISION-BASED FOLLOWING

The first technology demonstration, vision-based following, is ideal for close following of humans over an unknown route, or following where a GPS signal is not present. Vision-based robotic following using image segmentation and fuzzy-logic control has been previously demonstrated on a small research robot.[3] The method described here is an implementation of an adaptive mean-shift tracker that is similar to work done at the University of Massachusetts, Amherst, using the RMP.[4]

Before entering autonomous visual following mode, the system must first undergo an initialization stage. During initialization, the leader stands in front of the platform-mounted camera so that the camera is pointing directly at the leader's torso. The system will then visually "lock on" to the clothing covering the leader's torso, and enter autonomous following mode. Once in autonomous mode, the orientation and velocity of the Segway platform are adjusted so that the platform is a constant distance from the leader, and pointing directly at the leader. The robotic transporter follows a few meters behind a human leader, and responds immediately to the leader's movements, a behavior similar to that of a well-trained dog. The system depends entirely on visual characteristics of the leader's torso to track the leader.

3.1 Hardware Configuration

All computation and platform control is performed with hardware mounted on the RMP platform. A small color camera mounted to the RMP, and powered by a small battery, acquires the visual data which is fed to the visual following algorithm. The NTSC video is digitized with a small USB video digitizer that is attached to a laptop computer mounted on the RMP platform.

3.2 System Architecture

The visual following system uses two distinct algorithms. The visual tracking algorithm is responsible for maintaining a visual lock on the torso of the human leader. The output of this process is the calculated distance to the leader and the azimuth of the leader relative to the direction of travel of the Segway RMP. The second algorithm accepts the input from the visual tracker, and uses the information to direct the RMP platform to follow the leader. The algorithms are described in detail below.

3.2.1 Visual Tracking Algorithm

The visual tracking algorithm works by tracking a computational model of the clothing worn on the torso of the human leader. The hue of the clothing is captured in an initialization step during which the leader stands immediately in front of the camera for a few seconds upon system start-up. Alternatively, if a model to be tracked is known a priori, for example, the red cross on a medic, the initialization step can be skipped.

Once a model has been identified, the visual tracker tracks the size and centroid of a bounding box which surrounds the region. An example of the bounding box, centroid, and initialization area is shown in a screenshot from the user interface in Fig. 5. All calculations at this stage are performed in image coordinates (pixels). The movement of the centroid of the tracked area to the left or right of the center of the image indicates that the leader is moving to the left or right of the RMP platform. The area of the region enclosed by the bounding box is related to the distance of the leader from the camera. A growing bounding box indicates the RMP is approaching the leader, while a shrinking bounding box indicates the distance between the leader and the RMP is increasing.

The first step in visual tracking is the conversion of the incoming real-time digital video into the hue-saturation-value (HSV) color space. The HSV color space is preferable to the more common red-green-blue (RGB) space because the "color" information capture in the hue of the HSV space is independent of the saturation and brightness of the color. This makes the HSV space much more robust to varying lighting conditions.

The HSV color data from incoming video is fed to a continuously adaptive mean-shift (CAMSHIFT) algorithm. The implemented CAMSHIFT algorithm is similar to that developed at Intel for face recognition applications.[5] The CAMSHIFT tracker operates by modeling the object to be tracked as a color probability distribution. Color histograms in the HSV color space are used to form the color probability distributions.

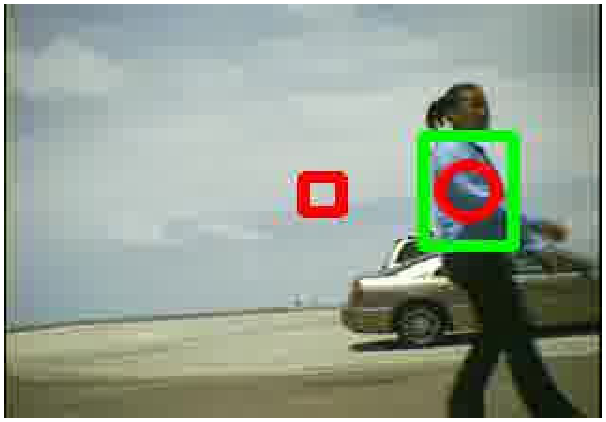


Figure 5. The square in the middle is the area used for initialization upon system start-up. The large rectangle is the bounding box surrounding the leader's torso. The circle in the interior of the rectangle is the centroid.

The tracker locates the best match to the model in incoming data by searching for the best fit within a search window in each incoming video frame. The size of this search window may be adjusted depending on the application. A non-parametric method for climbing density gradients is used to find the optimal match for the model within the search window.

Because the appearance of the leader may change over time due to changes in lighting and perspective, the model must be continuously adapted. The model representing the leader is recalculated during each iteration of the algorithm so that small changes in appearance are accommodated by the model. The update rate is typically 20-30 Hz.

The outputs of the visual tracking algorithm are two parameters: 1) the azimuth of the leader relative to the current heading of the RMP platform, and 2) the distance of the leader from the RMP platform. These two control parameters are sufficient to allow the RMP to follow a human at walking pace. Fig. 6 shows some sample output control data from the visual following algorithm.

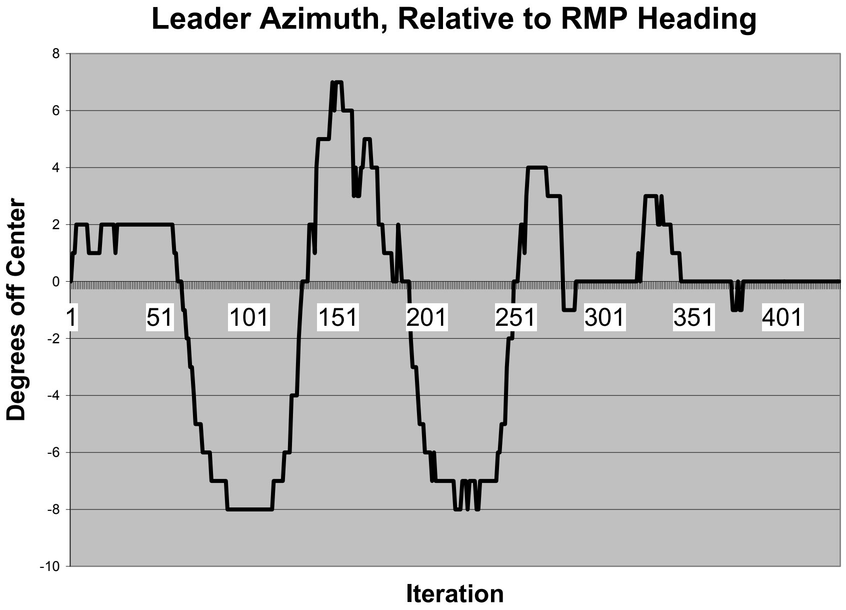
The CAMSHIFT algorithm is both accurate and fast. An only moderately optimized implementation of the algorithm, operating on a 640x480 image at 30Hz, occupied approximately 5% of the CPU time of a 1.6 GHz Intel Pentium 4 computer.

3.2.2 Visual Following Algorithm

Before being fed to a robot control algorithm, the incoming data is smoothed to filter out noise. This is particularly important with the somewhat noisy distance-to-leader data, as shown in Fig. 6. A simple adaptive mean filter operating on a 0.25 second window is used as the filter. Though this filter introduces a 0.25s lag in the reaction of the robot to the leader's movement, this lag has almost no noticeable effect at normal walking speeds.

The visual following algorithm employs simple proportional-integral-derivative (PID) loops for the control of both the turning and velocity of the RMP. The tuning parameters of the PID loops were obtained by experimentation. The PID loop controlling the heading of the platform simply operates by seeking to center the leader within the field-of-view of the camera. The farther off-center the leader is, the greater the turning response generated by the PID loop. Similarly, a PID loop controlling the RMP velocity seeks to maintain a constant distance from the leader. Velocity is increased as the distance increases in order to "catch up" to the leader.

Obstacle detection and obstacle avoidance (ODOA) is generally not needed in this close-following application because the platform travels immediately behind a leader, who, presumably, serves the role of an intelligent path planner, and avoids any objects which could cause problems for the platform. The PID controller prevents the robotic transporter from colliding with the human leader.



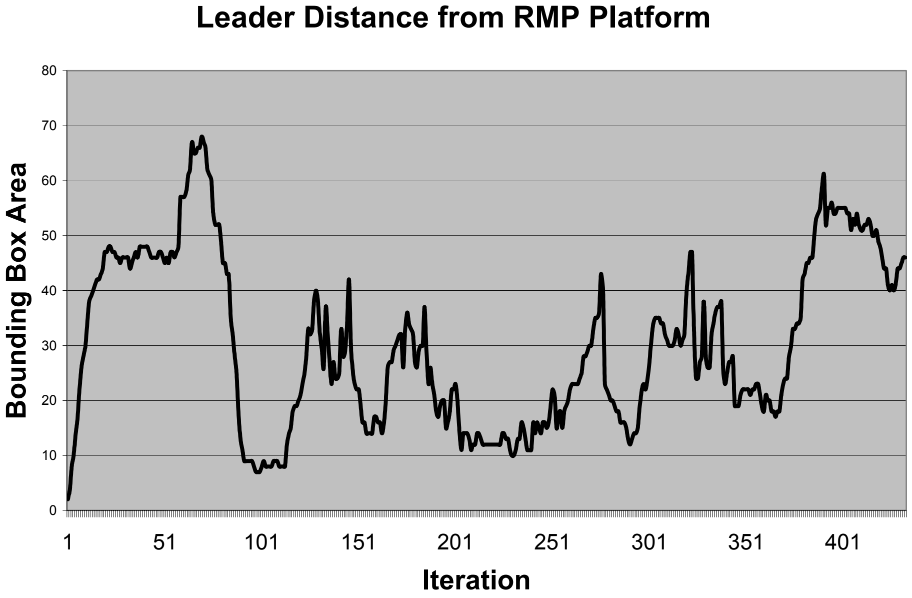


Figure 6. These charts were generated from input data showing the "leader" weaving from left to right while varying his velocity. The top chart shows the lateral movement of the leader, while the bottom chart demonstrates the distance from the camera. Note that distance data is considerably noisier than heading data. The data is smoothed before being fed to the robot controller. In this demonstration, the robot control algorithm has been turned off so that the data only represents visual tracking information**.**

3.3 Needed Improvements

There are a number of minor issues with the operation of the visual-following system as currently implemented. For example, if the leader suddenly turns sideways so that only his profile is exposed to the camera, the bounding box surrounding the tracked area will suddenly shrink, falsely appearing to the robot as if the leader is increasing his distance from the camera, and inducing an inappropriate response from the robot. The opposite problem occurs as the leader swings his/her arms. This increases the area of the bounding box, making the leader appear closer. In general, the distance measure generated from the use of bounding box area is considerably noisier than the heading data. An improvement could probably be made by using the height rather than the area of the bounding box. The use of a laser or sonar ranging sensor in addition to the visual data would also serve to correct most of these problems.

4. GPS-BASED FOLLOWING

In the second technology demonstration, the separation between leader and follower was significantly increased to several hundred meters using GPS information. This separation is currently limited only by the range of the 802.11b communication link between the two units. The track of the human leader, with a GPS unit in his/her backpack, was transmitted wirelessly to the robotic transporter (follower), which was also equipped with a GPS unit. The robotic transporter traced the path of the human leader by following these GPS breadcrumbs.

GPS-based robotic following is one of the near-term robotic efforts of the U.S. Army's Future Combat Systems (FCS), and has been the subject of demonstrations through the Demo III Experimental Unmanned Vehicle program.[6] Though similar, the GPS-based following technology presented here represents a simple solution for a smaller, slower class of robotic vehicles than that being developed for the FCS program.

4.1. Hardware Configuration

The principal components of this demonstration were a pair of NavCom Technology SF-2050G dual-channel GPS receivers with StarFire licenses. Instead of providing the logistics transport cart (follower) with local obstacle avoidance capabilities, we decided to simplify the solution and rely on more accurate GPS data. We depended on the leader to pick the path carefully, thus providing the obstacle avoidance function for the follower. This is accomplished using the StarFire GPS correction capability, an optional license that can be acquired for the NavCom GPS receivers. The StarFire system is a unique satellite-based wide-area differential GPS system providing better than 10cm horizontal accuracy anywhere on the earth's surface between latitudes 76°N and 76°S. The SF-2050 receivers use compact tri-band antennas capable of receiving both GPS and StarFire signals.

The robotic transporter's GPS receiver is augmented with a Microstrain 3DM-G Gyro Enhanced Orientation Sensor (electronic compass). This sensor provides initial directional information to the controller, since the GPS receiver cannot provide orientation information without having first moved some distance. Processing power is provided by two laptop computers, one in the leader's backpack, and the other mounted on the logistics cart. Communications between the two units are through 802.11b wireless modems (built into the leader's laptop, and as a PC Card with external range-extending antenna on the logistics cart).

4.2. Software Algorithm

The software for this demonstration involves two Kalman filters running on the leader and follower laptops, and communications using the Small Robot Technology (SMART) protocol, a robot communication protocol developed at SSC San Diego.[7] The Kalman filters and PID controllers described here were first developed in a prior GPS waypoint navigation project at SSC San Diego.[8] The SMART protocol is a robot communication and control protocol built on top of the UDP/IP protocol, and allows multiple robots and sensors to communicate peer-to-peer over a wireless network. SMART has several characteristics that make it ideal for robot communication:

- UDP offers lower-latency data transmission compared to TCP. This is particularly important during teleoperation of mobile robots. While UDP offers no error correction or transmission guarantees, SMART contains a simple, low overhead error-correction scheme which accounts for most types of transmission error.
- SMART is a peer-to-peer protocol. There is no central server. Every SMART "agent" contains a list of all other robots and devices, and initiate direct communications with any of them at any time. The protocol is also fault-tolerant—the failure of any one device does not affect any other SMART device.
- SMART offers a self-registering capability. Robots announce their existence to all other SMART agents via a broadcast message over the network. A "heartbeat" message from all SMART agents is used to maintain up-to-date lists.
- While most common robot control messages are supported by SMART, the protocol is also extensible, and can handle arbitrary messages and data types.

Fig. 7 summarizes the flow of data through the GPS-based following algorithm. At the start of a mission, the NavCom GPS units must first be placed at a site unobstructed by trees and buildings, and turned on for approximately 30 minutes. This allows the GPS to stabilize and obtain a good fix. At least four satellites must be in view at all times for reliable GPS data. The more satellites in view the better the accuracy of the GPS data and the less susceptible it is to erroneous jumps.

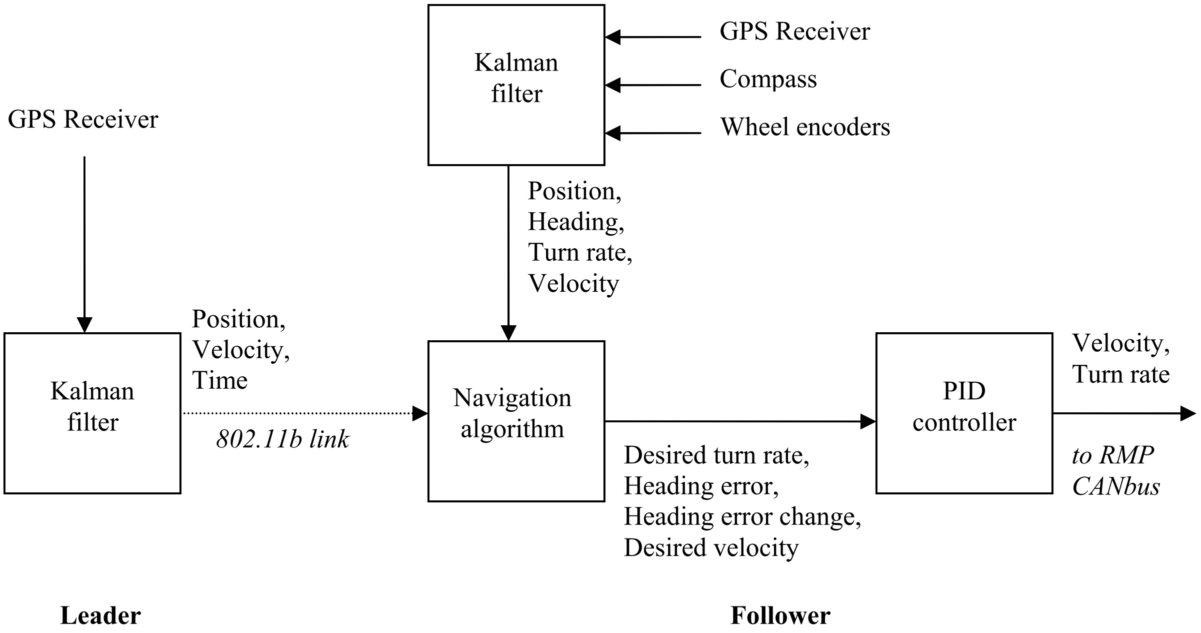


Figure 7. Functional diagram for the GPS-based leader/follower software.

The leader should start at least four meters away from the follower but still within communications range. Furthermore, there should initially be a straight, unobstructed path between the leader and the follower. This is because the leader normally picks the path to avoid obstacles, but there is no pre-defined path between the two at start up. Upon start up the follower will head straight for first waypoint (towards the leader)—see Fig. 8.

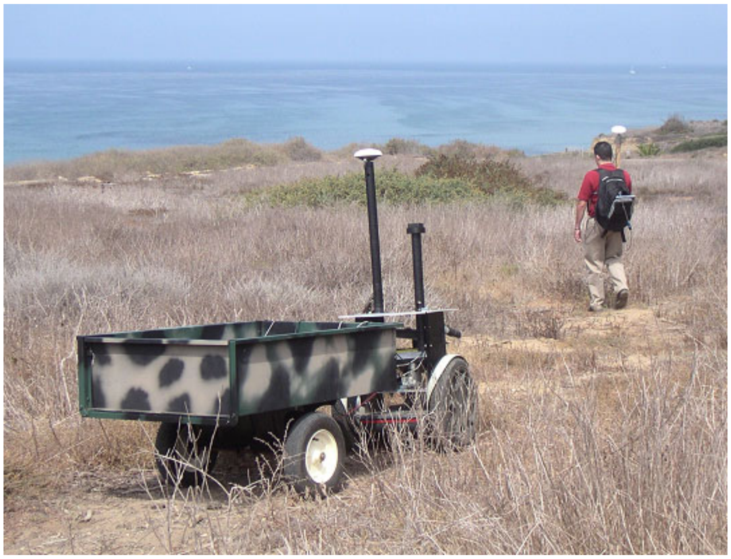


Figure 8. The leader and follower at the start of a GPS-breadcrumb following mission. The two masts mounted to the RMP host the GPS antenna and compass.

As the leader's controller code starts, an initial GPS lock and at least 10 good GPS data points are required before the Kalman filter starts filtering the data. The leader must be moving at a speed greater than 0.1 m/s for about 2 seconds to allow the GPS to obtain a good initial heading and velocity for the Kalman filter. The Kalman filter uses current and past GPS heading and velocity information to predict the position of the next GPS point. It then compares the predicted position with the actual GPS point received on the next cycle, making adjustments to both the actual data point and its internal states. The practical effect is the correction of outlying points that are caused by errors in the GPS signal.

After the leader's raw GPS data is smoothed by the Kalman filter, it is packaged into a SMART status data structure. This data structure is sent out over the wireless link to the follower using the SMART protocol. The structure contains the leader's position, velocity, and time stamp. GPS position is updated by the NavCom receiver five times per second. GPS heading and velocity are updated once per second. These update rates are the current limits of the NavCom GPS receiver. Updated Kalman-filtered status packets are sent out over the network five times per second.

The follower passes its own GPS, compass, and wheel encoder data five times per second to its Kalman filter to be smoothed. The Kalman-filtered GPS position, heading, velocity, and turn rate data are then used by a navigation algorithm, together with the GPS coordinate and velocity sent by the leader, to determine a command velocity and turn rate for the Segway RMP.

The navigation algorithm has a settable look-ahead parameter that determines how close the follower should try to approach the leader's past positions. A large look-ahead enables the follower to "look ahead" at several received breadcrumbs and generate waypoints that smooth out the path, cutting corners where reasonable. Since the follower currently does not have obstacle avoidance and depends on the leader to perform this function, we keep this parameter small, forcing the follower to pick waypoints that are close to the actual breadcrumbs, closely retracing the path of the leader.

The difference in heading (heading error) between the desired heading to the next waypoint and the follower's current heading is calculated in the navigation algorithm, as well as a heading error change (difference between current heading error and last heading error). The desired turn rate, heading error and heading error change are fed into a PID control loop to determine the turn rate needed to reach the goal point. The command velocity is set to the same as the leader's velocity up to a maximum of 1 m/s. The turn rate is limited to a maximum of 0.40 rad/s. As the follower's position and orientation data is updated, the turn rate is adjusted to keep it on track. As the Segway reaches one goal point, another goal point is calculated until the path end point is reached (i.e., there is no more data point in the receive queue of breadcrumbs). Fig. 9 shows the recorded path of the leader superimposed on the path of the follower for one of our demonstration runs. Figs. 10 and 11 show the heading error and distance between the leader and follower as functions of time.

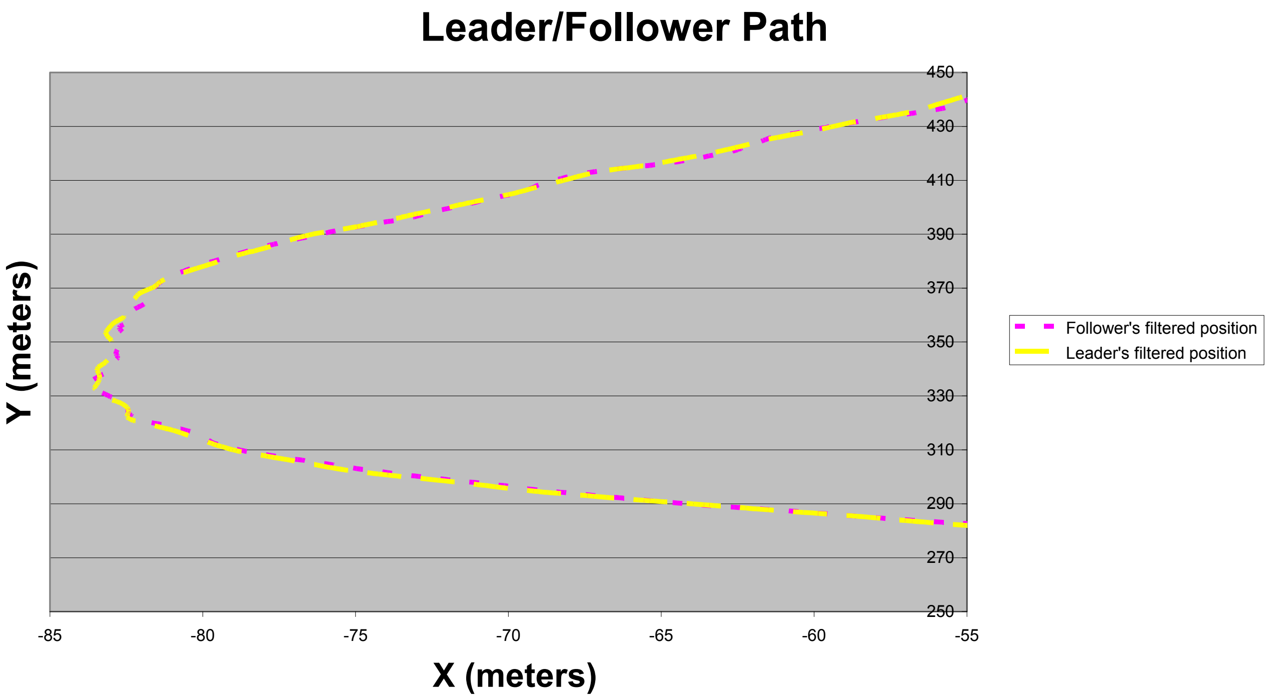


Figure 9. The Kalman filtered path of the leader (yellow, coarse dashes) overlaid over the path of the follower (pink, dashed).

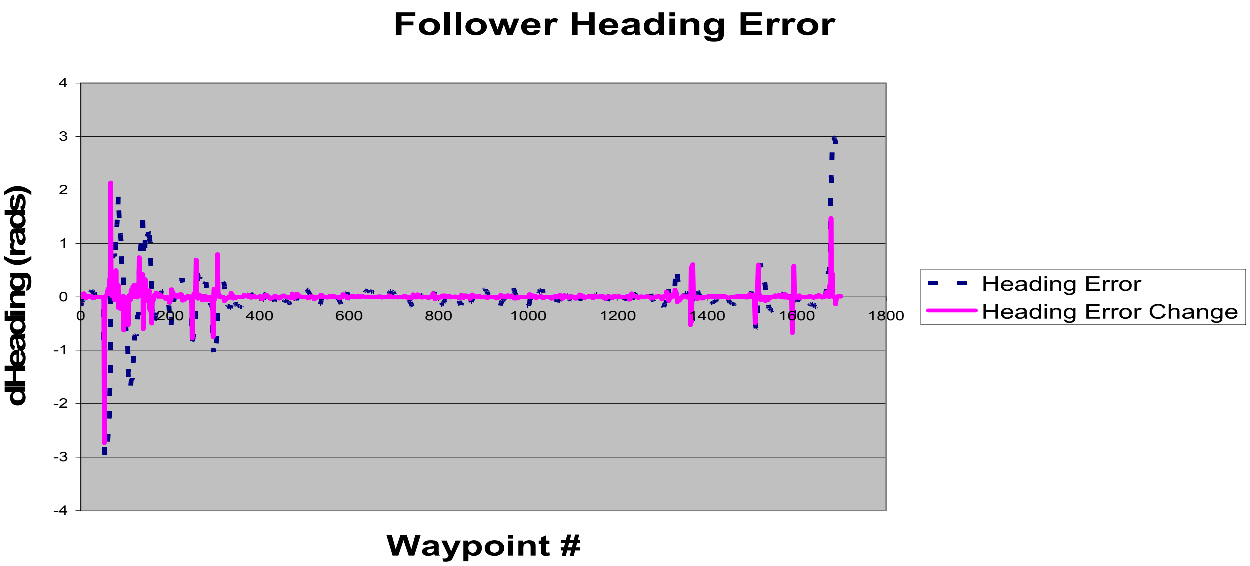


Figure 10. Heading error as a function of time (waypoint #). The large variance at the beginning is due to the settling of the Kalman filters.

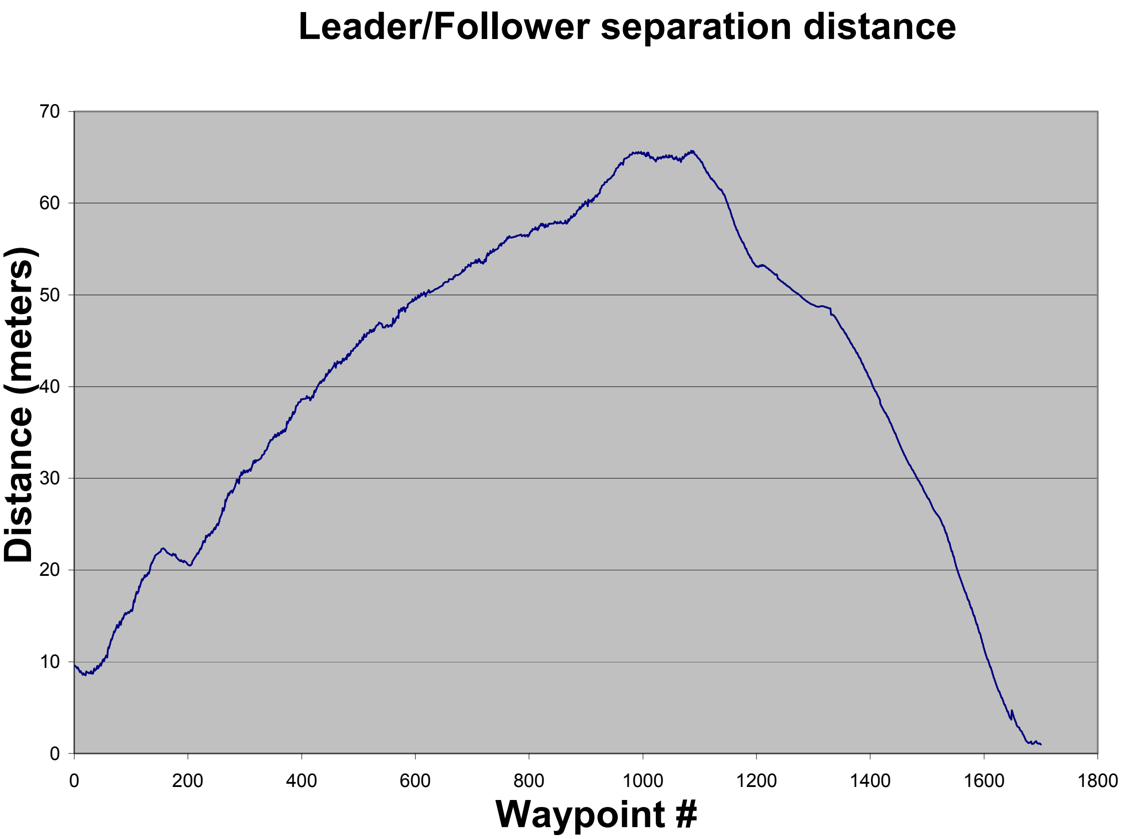


Figure 11. Distance between leader and follower as a function of time. The upper slope for the first half of the data is due to the speed ramp up by the leader, followed by the follower. Also, the leader was walking faster than a software-limited maximum speed setting on the follower. The downward slope occurs when the leader stopped, and the follower caught up.

4.3. Needed Improvements

The lessons learned and recommendations for improvements from this experiment include:

- Current turn rate is limited in software to 0.40 rad/s because of the simplistic design of the connection between the trailer and the Segway RMP. The bar connecting the two is jammed against one of the Segway RMP's fenders on sharp turns. This creates a run-away condition, as the Segway RMP keeps trying but unable to gain a certain attitude. The limited turn rate reduces mobility and reaction time. A better mechanical design that eliminates this problem but still keeps the hitch location near the wheel axis is needed.
- The 3DM-G compass needs time to stabilize to new headings and cannot react to quick turns. Furthermore, both the Segway RMP's internal gyroscope and the 3DM-G have fairly poor accuracy, forcing us to use the RMP's wheel encoder data to calculate turn rate when the follower is moving. The RMP's wheel velocity encoders are stable and show very little variance. However, when wheel slippage occurs (e.g., on a sandy surface or gravel road), the Segway's orientation data is thrown off, and this in turn throws off the Kalman filter and subsequent dead reckoning. A better gyro and compass (or especially the University of Michigan's FLEXnav unit)[9] would significantly improve navigation.

5. CONCLUSION

Capitalizing on the small size, ruggedness, and large payload capacity of the new Segway Robotic Mobility Platform, we have conducted several demonstrations of military-applicable robotic transportation on an organic (personal) level. These include providing roboticized mobility for the Life Support for Trauma and Transport unit, and leader/follower applications using computer vision and GPS-based techniques. These were simple individual technology demonstrations. To provide a complete, robust system, additional capabilities would be needed. These may include an obstacle avoidance capability based on ladar or stereo vision, and a more capable Inertial Measurement/Dead Reckoning System.

ACKNOWLEDGMENTS

This work was supported by the Defense Advanced Research Projects Agency (DARPA) Information Processing Technology Office (IPTO) as part of the Mobile Autonomous Robot Software (MARS) program.

REFERENCES

1. H. G. Nguyen, J. Morrell, K. Mullens, A. Burmeister, S. Miles, N. Farrington, K. Thomas, and D. Gage, "Segway Robotic Mobility Platform," *SPIE Proc. 5609: Mobile Robots XVII,* Philadelphia, PA, October 2004.
2. Integrated Medical Systems, Inc., *LSTAT,* URL: http://www.lstat.com/lstat.html.
3. M. Tarokh, "Robotic person following using fuzzy control and image segmentation," *Journal of Robotic Systems*, Vol. 20, No. 9, pp. 557-568, September 2003.
4. University of Massachusetts, Amherst, *UMass Segway,* URL: http://www-robotics.cs.umass.edu/segway/.
5. G. R. Bradski, "Computer video face tracking for use in a perceptual user interface," *Intel Technology Journal*, Q2 1998.
6. B.E. Brendle, Jr. and J. J. Jaczkowski, "Robotic Follower: near-term autonomy for future combat systems," *SPIE Proc. 4715: Unmanned Ground Vehicle Technology IV*, Orlando, FL, April 2001.
7. Denewiler, T.A. and R.T. Laird, "NERD: Network Enabled Resource Device," *AUVSI Unmanned Systems 2002,* Lake Buena Vista, FL, July 9-11, 2002.
8. Bruch, M.H., Gilbreath, G.A., Muelhauser, J.W.,and J.Q. Lum, "Accurate Waypoint Navigation Using Non-differential GPS*," AUVSI Unmanned Systems 2002,* Lake Buena Vista, FL, July 9-11, 2002.
9. L. Ojeda, M. Raju, and J. Borenstein, "FLEXnav: A Fuzzy Logic Expert Dead-reckoning System for the Segway RMP." *SPIE Proc. 5422: Unmanned Ground Vehicle Technology VI,* Orlando FL, April 2004.