Obstacle Avoidance with a Segway RMP

Nathan Wiedenman
Department of Civil and Mechanical Engineering
U.S. Military Academy, West Point, New York

Raymond Von Wahlde
Weapons Technology Analysis Branch
Weapons and Material Research Directorate
U.S. Army Research Laboratory
Aberdeen Proving Ground, Maryland

Wesley Brown
United States Corps of Cadets
United States Military Academy
West Point, New York

1 Executive Summary

Robotic ground and aerial vehicles are becoming more of a vital part of the U.S. Army's modern warfighting capability every day. The ever-increasing load on the individual infantry soldier has prompted several studies into creating a semi-autonomous "mule" vehicle to relieve soldiers of some of their load.

This research seeks to design such a mule using a variant of an off-the-shelf Segway Human Transporter (HT) vehicle. This vehicle has built-in control algorithms which enable it to balance independently, and to accept steering input from a human passenger. The Robotic Mobility Platform (RMP) is a variant of the HT which balances independently and accepts steering information through serial ports.

This paper addresses the development of an obstacle avoidance capability in the Segway RMP. This capability was realized using a Sick scanning laser and a novel path-seeking algorithm implemented using LabVIEW software.

Initial results for obstacle avoidance are discussed. In addition, plans for future work to create a follower capability in the RMP are delineated.

2 Introduction

The United States Army has been vigorously pursuing robotic solutions to several problems. Increasingly, robotic vehicles and platforms provide reliable intelligence input and hazard identification for Soldiers in the field and in combat zones around the world. One problem that remains to be adequately solved is the need for a semi-autonomous follower vehicle, or "mule," that will enable ground troops to have ready access to equipment and supplies without needing to physically carry the weight themselves.

2.1 Background

Robotic vehicles offer an attractive option for performing dangerous or monotonous missions in the place of a human Soldier. For example, robotic vehicles are currently being used in Iraq to examine and identify improvised explosive devices (IED). This allows the Soldier to see and possibly disarm an IED without ever having to physically approach it (see Figure 1) [1].

Figure 1. A robotic vehicle is used in Iraq to examine a suspected IED from a safe distance [1].

The load on the individual infantry Soldier has continued to increase despite efforts to limit it. Initiatives to better equip, arm, and protect the Soldier have left him weighed down with ammunition, radios, body armor, individual weapon, anti-tank weapons, grenades, food, water, and so on. This weight not only makes it difficult for the Soldier to perform his mission, it makes it difficult at times for him to even move. As a result, the Army has begun to investigate possible alternatives for transporting the Soldier's load.

The most promising alternative is the mule concept. A robotic vehicle (the mule) will travel with a small element of Soldiers, carrying heavy equipment for ready access. The mule remains a specified distance from the unit, so as not to reveal the unit's position. It will travel semi-autonomously, making its way around obstacles and over rough terrain without direct

involvement by the Soldiers. It recognizes its position relative to the unit, and will move to a specified location on command.

2.2 Selection of the Segway RMP

The Segway Robotic Mobility Platform (RMP) offers several unique characteristics as a robotic vehicle. The RMP is a variant of the Segway Human Transporter (HT), a commercial vehicle for individual transport. Each of these vehicles is essentially a large inverted pendulum, requiring constant control input to maintain an appropriate position (see Figure 2).

Figure 2. Segway HT (left) [2] and Segway RMP (right) [3]

These vehicles move utilizing differential drive locomotion. This is a mechanically simple system that requires rather complicated controls to enable it to move in reliably accurate path [4]. All of these controls are internal to the Segway vehicles. Normally, a two-wheeled differential drive robot requires an additional caster wheel to maintain balance. The HT and RMP maintain their balance by means of an internal control algorithm. That is, the user does not need to actively participate in the balance mechanism of the vehicle. Left untouched, both vehicles will balance autonomously. The RMP therefore requires only movement commands.

This autonomous balance capability enables a vehicle geometry that provides extremely good maneuverability under certain conditions. The RMP has a zero turning radius and can reach a maximum velocity of 3.54 m/s (8 mph) [3]. The vehicle also exhibits great ease in reversing direction and negotiating a nonlinear course.

The RMP was seen as a good possible mule candidate due to its ability to carry a 45 kg (100 pound) payload.

3 Implementing Obstacle Avoidance

3.1 Movement Characteristics

Due to the inverted pendulum nature of the RMP, the vehicle has unique movement characteristics. All fore and aft movements of the RMP are implemented by causing the vehicle's center of gravity to fall in the direction of desired movement, and then chasing the CG to prevent falling. In order to move forward, the RMP must first move slightly backwards in order to "tip" itself in the direction of desired movement. Similarly, the vehicle must accelerate slightly in order to stop, in effect tipping itself away from the direction of motion. This causes slight delays in execution of desired movement commands. As a result, a safety buffer must be included in the command instructions to account for the extra room needed to start and stop.

These delays in movement are illustrated in Figure 3. This figure shows the response of the RMP to a step velocity input. The vehicle is commanded to begin from rest and move at 1.77 m/s (4 mph) forward. The allowable acceleration was set to one half of maximum [3].

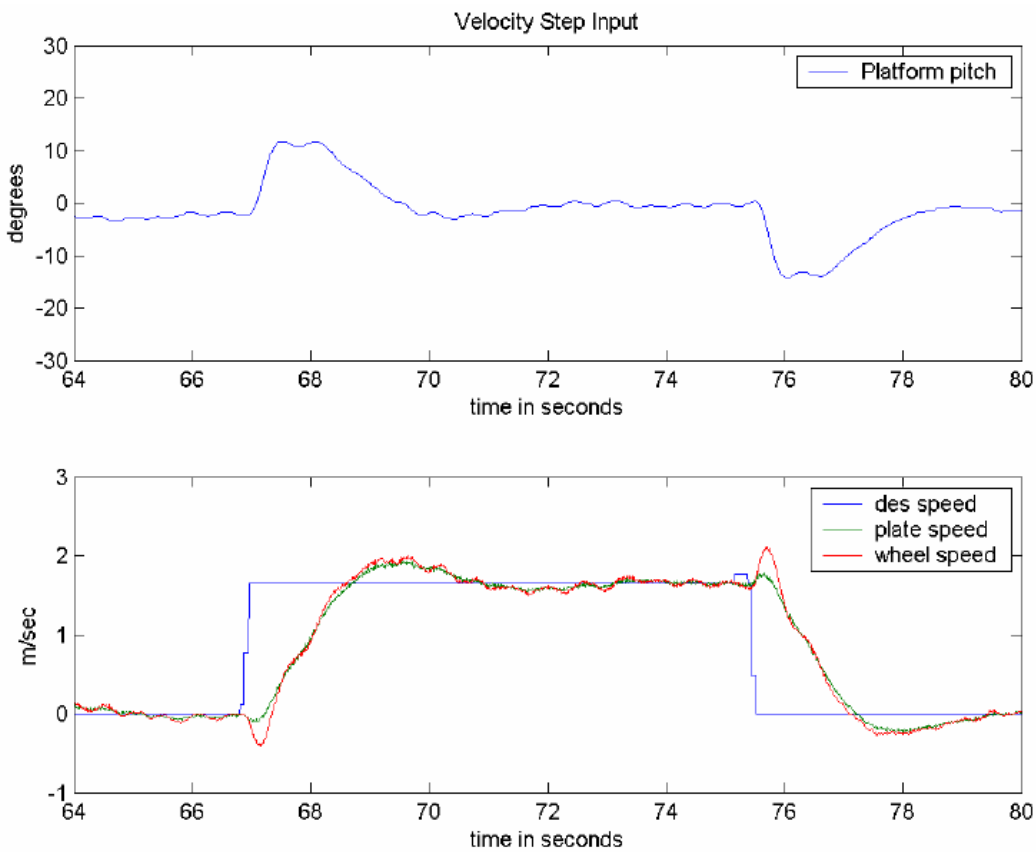

Figure 3. RMP response to velocity step input [3].

The first plot shows the pitch of the vehicle's top platform during the step response, with positive defined in the direction of desired motion. It can easily be seen that the vehicle oscillates slightly at all times in order to maintain its balance. When the movement command is received, the vehicle accelerates briefly in the opposite direction, causing the top plate to pitch in the direction of desired motion. The wheels then move the vehicle forward to keep the vehicle from falling. A similar process occurs when the vehicle is commanded to stop. The wheels accelerate briefly in the current direction of motion, outrunning the vehicle CG and reducing speed. The vehicle then reestablishes equilibrium [3].

Safety was an overriding concern during all of our work with the RMP. The vehicle has powerful motors, and the internal balancing algorithm can cause the vehicle to move in a non-intuitive fashion at times. For example, when the RMP is powered on it automatically finds its balance and then maintains it. If the vehicle is tipped in any fashion, it will drive itself in such a manner as will enable it to regain its balance. This may cause the RMP to accelerate rapidly and dangerously unless the source of the tipping input is removed.

If the RMP is unable to reestablish a balanced position, it will automatically shut itself down. This of course results in the vehicle falling to the ground, a situation which could easily damage any sensor package or sensitive payload on the upper platform. To prevent such potentially catastrophic impacts, CDT Brown fashioned legs and attached them to the RMP to limit the extent of any fall (see Figure 4).

Figure 4. RMP with fall-limiting legs attached. The emergency stop tethers are the cords hanging from either side of the RMP, with clips attached at the ends.

The final safety measure utilized was an emergency stop. There were two quick-disconnect switches connected to tethers which allowed an operator to shut down the RMP without physically contacting the moving vehicle. These emergency stop tethers can be seen in Figure 4.

3.2 Communication

The first challenge encountered was properly interfacing with the RMP in order to provide it the necessary movement commands. Instructions are sent to the RMP through a pair of serial ports on the underside of the electronics box. A laptop communicates with the RMP through these serial ports by means of two Controller Area Network (CAN) cards. Demo software was available which would allow us to communicate with the RMP. In addition, we needed to be able to "drive" the vehicle using a USB joystick so that we had more direct control of movement when needed. This setup is illustrated in Figure 5.

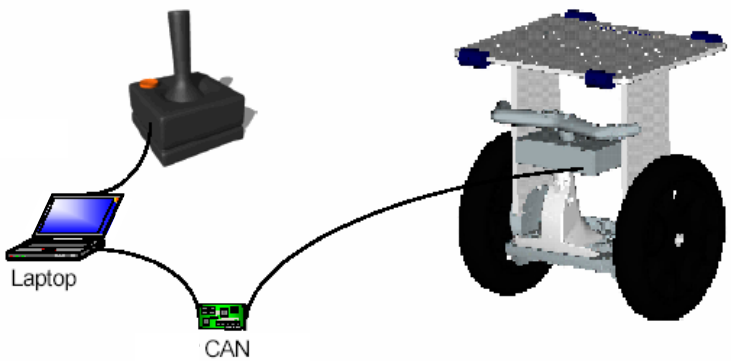

Figure 5. Serial communication with the RMP [3].

The next challenge faced was interfacing with a sensor package. A Sick LMI 4000 laser was chosen for ease of use and reliability. Obstacle data was obtained from the laser in the form of hexadecimal data through a serial port connection. Using the Sick software, a 180° profile of the obstacles was easily generated.

The primary concern at this point was to integrate the obstacle information with the movement command information efficiently so that the vehicle could move autonomously. We decided to use LabVIEW to provide a common interface and connectivity between the sensor (laser), brain (laptop), and vehicle (RMP).

LabVIEW is a program that enables the user to collect external data and provide instructions to external equipment through graphical interfaces called virtual instruments (VI). This enabled us to use the same software platform to receive obstacle data from the laser (in the form of a hexadecimal array), interpret that data through an algorithm to determine path, and then send movement commands to the RMP via the CAN cards (see Figures 6 and 7). LabVIEW also enabled the direct use of the USB joystick to provide manual movement commands.

Figure 6. Testing the LabVIEW/laser interface for proper movement command output from the laptop. Note the obstacle profile shown on the laptop screen.

Figure 7. CDT Brown programming the RMP using LabVIEW. The Sick laser is the blue and black box mounted on the left of the RMP platform.

3.3 Obstacle Avoidance Algorithm

Using LabVIEW, we were able to implement a fairly simple, though robust, algorithm to enable the RMP to avoid obstacles encountered during movement. The first step was interpretation of the obstacle data coming from the laser.

The Sick LMI-4000 laser transmits a constant flow of obstacle information on command. LabVIEW was programmed to send out the initialization command prior to the start of movement. The incoming data was in the form of a hexadecimal array, containing information

about azimuth and distance. The laser scans a two-dimensional 180° fan by firing a beam every half-degree. It then senses how far each beam traveled before interruption, returning a beam index and distance for all 362 beams (0° to 180° by 0.5° increments) for each scan.

This laser data was constantly interpreted by the LabVIEW VI to generate an obstacle array. This array was a binary matrix, with 1 representing an open path and 0 representing an obstructed path along each of the beams. This array was converted into graphical form for ease of interpretation during testing.

A picture of a typical obstacle array is shown in Figure 8. The 362 beams emitted by the laser are plotted across the horizontal axis, numbered in the same fashion as the laser data (beam index 0 represents the rightmost beam, and index 361 represents the leftmost). Range in 100mm increments is plotted on the vertical axis, from 0m to 8m.

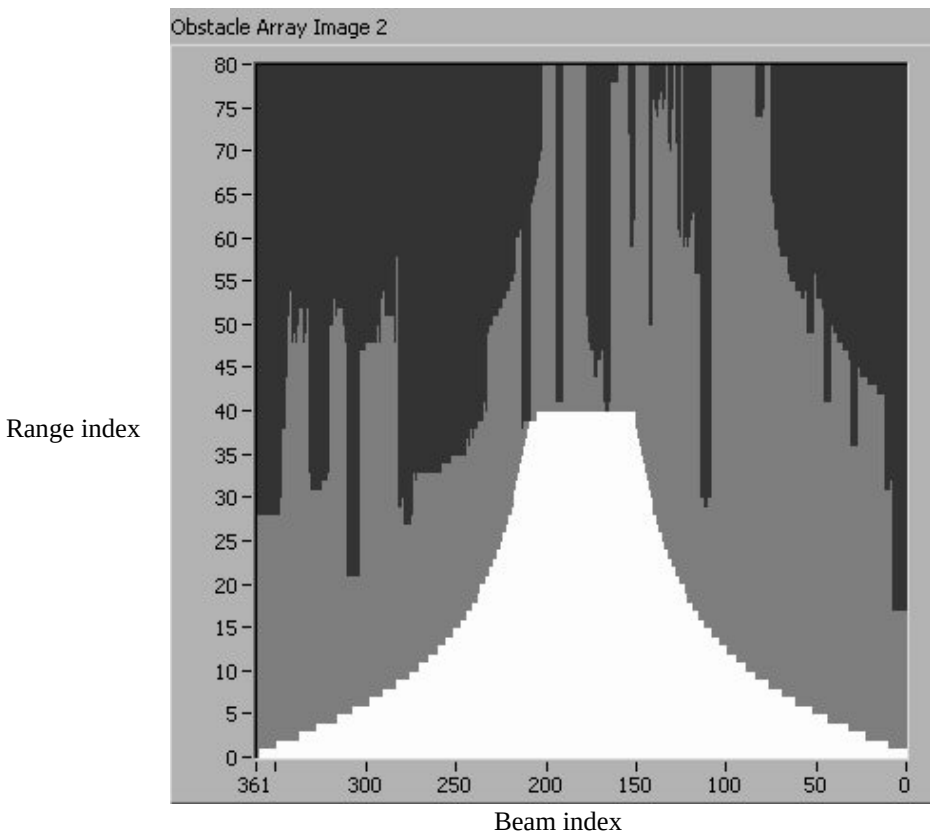

Figure 8. Obstacle array for a largest open area found at the center beam index. RMP moves straight ahead in this case.

Within the obstacle array, black regions represent obstructed beam paths, while gray and white represent unobstructed beam paths. The algorithm converts the dimensions of the vehicle into an equivalent number of beams for each range, shown in white in Figure 8. Because the

algorithm is converting radial data to a rectangular array, the white search area is wider at shorter ranges.

The algorithm searches for a path along each beam index from 0 to 361. That is, the program searches an area similar in shape to the white region in Figure 8 for each beam index, centered on that beam. As can be seen in Figure 9, the search areas for off-center beam indices do not include shorter ranges, as these shorter ranges do not include enough beams to represent the full dimensions of the vehicle.

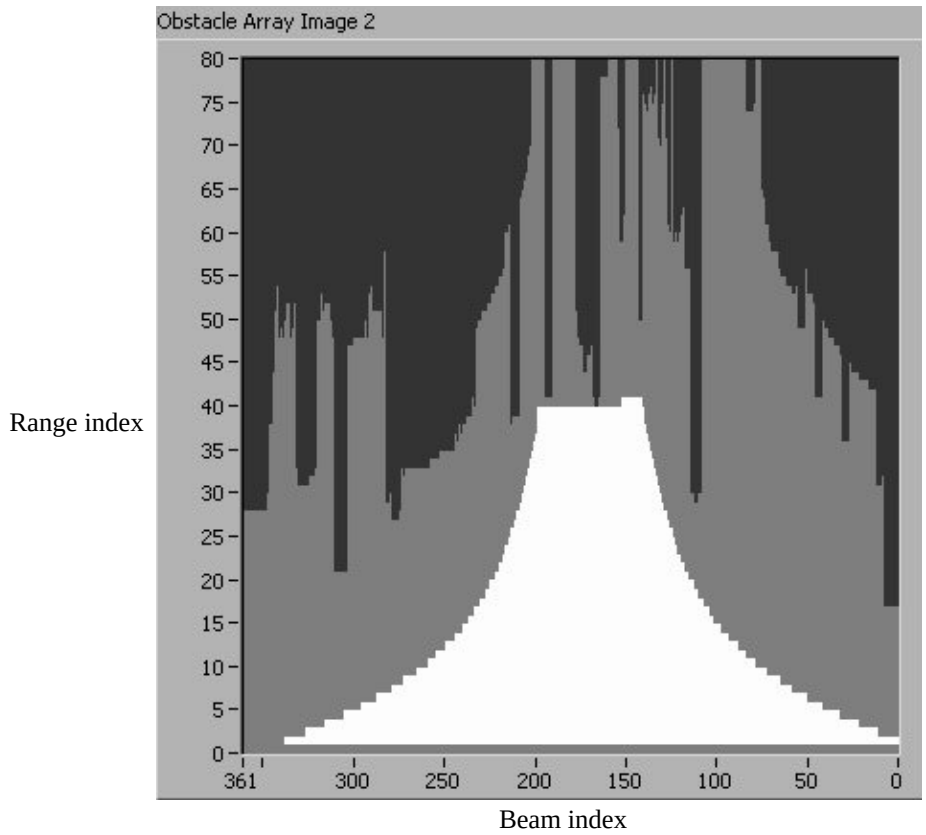

Figure 9. Obstacle array for largest open area found along beam index 169 (84.5°). The RMP will turn to the right by 5.5° before moving forward.

When the search encounters an obstacle, it stops searching for paths along that index. The total unobstructed area (the total number of 1's in the obstacle array that fall within the white search area) is added up. Included in this count for off-center beam indices is the area below the first range index searched. This rectangular area, while not wide enough to accommodate the vehicle, is clear of obstacles. If this area were not counted, the algorithm would be heavily weighted to paths in the center of the scan. As it is, the algorithm tends to favor center paths slightly over off-center paths.

Once the search is complete, the values for open area corresponding with each beam index are compared. If the largest open area is found within a small envelope around the center beam, the vehicle is commanded to move straight ahead. For searches that result in the largest

open area found outside of the center envelope, the vehicle turns to center itself on the beam index with the largest open area, and then begins to move straight ahead.

3.4 Performance

The RMP performed well when faced with an obstacle field within the sensing range of the laser. Since the laser was mounted to the top of the RMP platform (see Figure 7), obstacles shorter than approximately 750mm were not detected by the sensor.

On dry flat terrain, the RMP performed extremely well. The obstacle avoidance algorithm was initially tested indoors on a concrete floor. This provided good traction for the vehicle, allowing us to examine the performance of the algorithm separately from the vehicle's terrain capabilities.

The indoor testing showed that the vehicle could easily sense and react to obstacles encountered (see Figure 10). The algorithm was efficient enough to constantly interpret obstacle data, and as a result the vehicle could quickly react to dynamic obstacles. This is dependent upon the speed and acceleration settings of the RMP, which were set near 25% for the purposes of this test. Higher settings will require larger stopping distances, and thus larger buffer zones around the vehicle.

Figure 10. Indoor testing of the RMP.

When taken outside, some limitations of the RMP began to manifest themselves. Due to the nature of the vehicle's movement characteristics, the RMP does not easily surmount obstacles larger than 100mm. The vehicle will generally lose traction, and thus its ability to control its own movement. This causes rapid wheel acceleration and shutdown.

The RMP performed well on both asphalt and slightly rough terrain. It had no trouble negotiating a grassy field with gradual rises and small (<100mm) obstacles (see Figures 11 and

12). The performance of the obstacle avoidance algorithm showed no difference between indoors or outdoors.

Figure 11. Outdoor testing of the RMP (asphalt).

Figure 12. Outdoor testing of the RMP (grass).

The two-wheeled geometry of the RMP allowed it to maneuver very well when presented with tight paths. The vehicles ability to turn completely around within its own footprint allows it to get out of dead-end paths relatively easily.

The balancing programming internal to the RMP also caused trouble on rough terrain. Since the wheels are constantly moving, providing balance even when the vehicle is stationary, when one wheel loses contact with the ground the vehicle will shut down. More robustness in this feature is necessary.

3.5 Future Work

In order for the RMP to be used as a mule, it must have the capability to follow a designated master. This can be implemented using a radio link between the master and mule, with some small program modifications to tell the RMP what position to maintain relative to the master. In addition, a GPS capability will allow the RMP to move around obstacles and seek the best path back to the master.

To better overcome the shortcomings of the RMP's movement in rough terrain, map data for a given piece of terrain can be included in the laptop. This data can be interpreted to help the RMP avoid drop-offs and larger obstacles.

4 Conclusion

This work demonstrated the practical implementation of a novel and simple algorithm for obstacle avoidance using a Segway RMP vehicle. The algorithm worked well for obstacles within the viewing range of the sensor.

The RMP performed well under certain conditions. The geometry of the vehicle allowed it to maneuver within very tight constraints, enabling it to extricate itself from difficult positions with relative ease.

Several disadvantages of the RMP were identified in this work. The vehicle does not operate well on very rough terrain (obstacles larger than 100mm). The automatic shutoff features of the internal balancing software could be a drawback on rough terrain, as the vehicle will not operate with one wheel off the ground for even small periods of time.

References

[1] Hasenauer, Heike, "Faster Equipping of U.S. Forces," *Soldiers Magazine*, Vol. 59, No. 10, October 2004, pp. 20-23.

[2] Segway website, *www.segway.com*, Accessed 12 October 2004.

[3] *Segway Robotic Mobility Platform (RMP) User Instructions*, Version 1.3, Segway LLC, 2004.

[4] Jones, Joseph L., and Flynn, Anita M., *Mobile Robots – Inspiration to Implementation*, A K Peters, Wellesley, MA, 1993.