

Evaluating Path Tracker Performance for Outdoor Mobile Robots

S. A. Roth¹ and P. H. Batavia²

ABSTRACT

Accurate, high precision path tracking is an important capability for mobile robots in certain applications. In this paper, we discuss a formal method for comparing and evaluating the performance of path tracking algorithms. This method provides a way to quantify both the stability and performance of a tracker, and is of particular interest when high-accuracy off-road tracking is required. Previous work on path tracker performance has focused on the average and maximum position error over the entire path, or has dealt with simulated results. Overall, there is little analysis of tracking error at a lower level, which can provide insight into the behavior of the tracker, along with overall weaknesses and strengths. Our method includes measuring the tracker's response to particular situations such as discontinuities in the path's x, y coordinates and path curvature. We also examine the time for the tracker to settle onto the path and the distribution of tracking error vs. path curvature. We apply this method to two commonly used path trackers operating on an outdoor mobile robot. The first is based on a pure-pursuit model, and the second is based on work by Zhang et al (1997). By using this method we were able to evaluate the performance of these two trackers, and discover their relative strengths and weaknesses. The insight gained through this analysis has led to recommendations for path tracker development that will be presented.

KEYWORDS. Path tracking, Performance tests, Pure pursuit, Automation

INTRODUCTION

High accuracy path tracking is very important in certain applications. Indoor automated guided vehicles or vehicles navigating crowded environments need accurate path tracking to avoid collisions with obstacles. Precision turf management also requires accurate path tracking. Whether the goal is to mow or spread fertilizer over the turf, accurate path tracking increases process efficiency. Mowing a golf course or sports field requires accurate localization and path tracking to enable mowing patterns on the turf. Oscillations in the path must be kept below 5 cm to avoid visually obvious errors in the mowed pattern. While spreading fertilizer, too much or too little overlap in the coverage pattern leads to over or under fertilization. Therefore, measuring path tracking performance is important.

Previous work on path tracker performance has focused on the average and maximum position error over the entire path, or has dealt with simulated results. Baltes (2000) describes a benchmark suite for mobile robots that includes a performance test for path trackers. To measure a tracker's performance, the test uses average and maximum tracking error and also the timing error in the trajectory. Shin et al (1991) and Mouri (1998) describe tests performed on an actual robot, but with little formal analysis of the tracking error beyond visually comparing the reference and actual paths. Weigou et al. (1999) present tracking tests on straight and curved paths, but the results are from simulation. Garbutt et al. (2000), Petrov (1991) and Zhang et al. (1997) present tests from simulation. Overall, there is little analysis of tracking error at a lower level that can bring insight into the behavior of the tracker, along with overall weaknesses and strengths.

¹ stephan+@cs.cmu.edu

National Robotics Engineering Consortium, 10 40th st, Pittsburgh, Pa 15201

² parag@ri.cmu.edu

National Robotics Engineering Consortium, 10 40th st, Pittsburgh, Pa 15201

For linear time invariant systems there are tests available to determine a system's stability and settling time. One can calculate the system's transfer function's poles and zeroes, and from their locations determine if the system is stable and how quickly it will settle into equilibrium. However, most mobile robot systems are not linear time invariant, and can be subject to non-holonomic constraints. Therefore an empirical approach must be taken to measure the system's performance.

This paper presents a series of tests that can measure and compare the performance of path trackers on mobile robots. The tests measure the tracker's response to paths with step discontinuities, different curvature paths and a 'real world' path with curved and straight sections. The tracker's response to these paths can be used to measure the tracker's settling time, stability, tracking error versus curvature and overall path tracking error. The results of these tests can be used to compare the performance of two different tracking algorithms or while adjusting the control parameters in a single path tracker.

We apply these tests to two path trackers running on an outdoor mobile robot. The first tracker is pure pursuit, a common path tracker. The second tracker is based upon work by Zhang et al. (1997). By using these tests we are able to evaluate their performances and compare the relative strengths and weaknesses of the two trackers. Overall, both trackers have similar performance characteristics, but the Zhang tracker holds a slight advantage due to a faster settling time and better stability.

MEASURING TRACKER PERFORMANCE

Important Characteristics

When we speak of path tracking performance, we are most concerned with three characteristics. First is stability. It is important that a path tracker be stable when encountering moderate offsets from the desired path. A tracker must be stable when following different path types and at different speeds. It is most important to check stability at high speeds because vehicle and control system dynamics have a greater impact than at lower speeds.

A second characteristic of a path tracker is settling time. Settling time is the time required for a tracker to settle onto a desired path and stop oscillating given an initial offset from the path. When examining settling time we also want to know if the system is over damped, under damped or close to critically damped.

A third characteristic is overall tracking error. Tracking error is defined as the shortest distance between the robot's control point and the desired path. To show on which side of the path the robot is, the tracking error is negative when the robot is to the left of the path. When examining the tracking error, we want to determine how close the tracker stays to the path. We also want to measure the magnitude of any oscillations.

To measure these three characteristics, we perform tests using three different path types. The first test uses a path with a step discontinuity. The second test uses circular paths with different radii, and the third test uses a 'real world' path containing straight and curved sections.

Step Discontinuity Test

The step discontinuity test measures the tracker's stability, settling time and oscillation frequency (fig 1a). The initial straight section is used to get the robot up to speed, and also to get a baseline measurement of the straight line tracking error. The settling time and oscillation frequency are measured by treating the system as a damped oscillator. Looking at the actual path from figure 1a, the system is obviously acting as a damped oscillator. The tracking error vs. time is fit to an exponentially decaying cosine. This test is run at different speeds to ensure tracker stability. Although this is not a path a robot would be expected to follow in normal use, it does test the tracker's response to some real world situations. For example, when coming out of a turn there can be an offset from which the tracker must recover. If the robot's positioning system uses GPS, beacons or some other system that is prone to errors, this test simulates the tracker's response to

a discontinuity in the positioning system's estimate. The sizes of these offsets should be considered when choosing the size of step in this test.

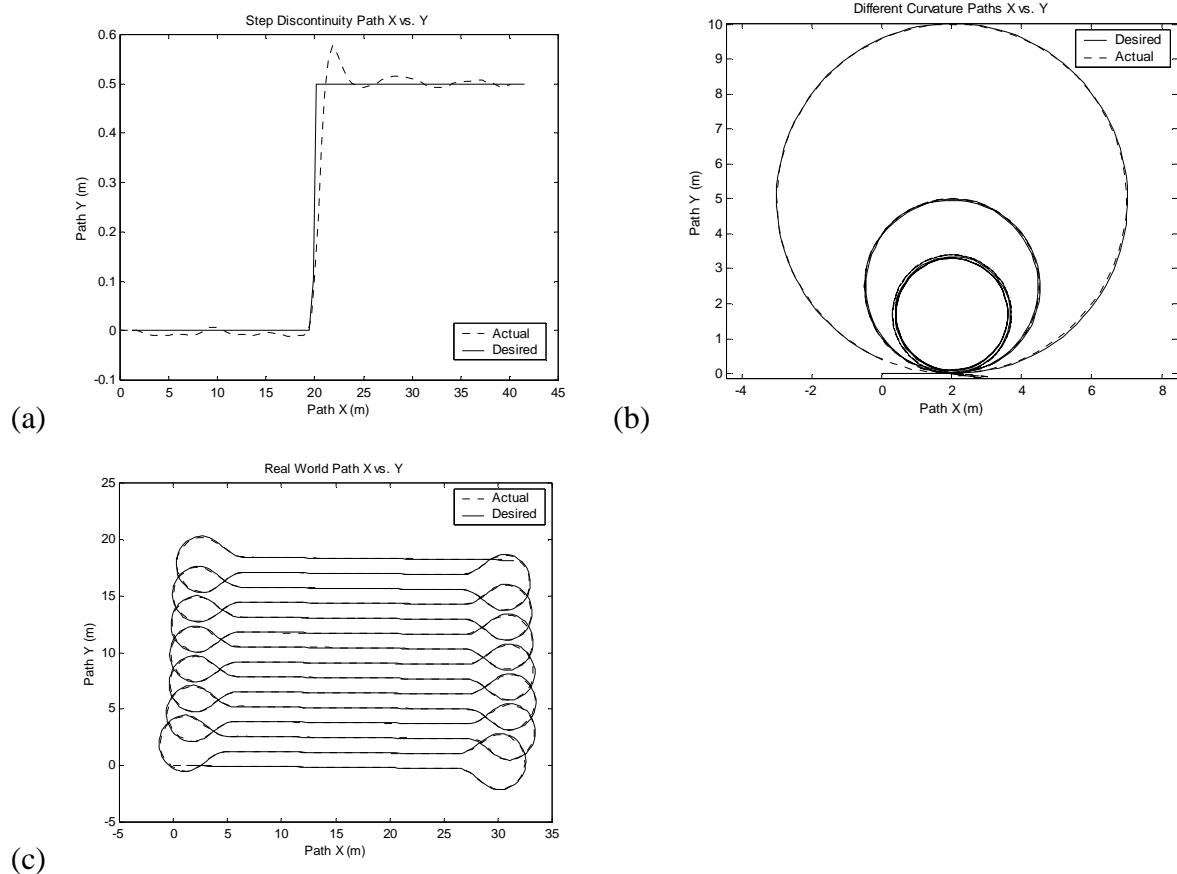


Figure 1: This figure shows the different path types used in the tracker tests. (a) Represents a result of the step discontinuity test. The axis scales are not equal to show the effects of the step discontinuity. (b) Shows the three circular paths used in the curvature test. The three paths are shown on the same graph for convenience. (c) Shows the path used in the ‘real world’ test.

Curvature Test

The second test uses circular paths of different radii (fig 1b) to measure tracking error as a function of path curvature. This test is run at a constant speed, and gives an indication of how well the tracker will follow paths of different curvatures. The range of curvatures should be from a straight line to the minimum turning radius the robot can or is expected to follow. One must make sure that the tracker does not constantly saturate the steering when following the smallest radius path. The test measures the steady state tracking error vs. curvature, and on the smaller radius paths it may be necessary to go around several times before the tracker reaches a steady state.

Real World Test

The third test is run on a path with a combination of straight and curved sections. This measures the tracker's performance on a ‘real world’ path. This test is used to measure the distribution of tracking error over straight and curved sections (fig 1c). Studying the distribution of tracking error over individual path segment types provides insight into the tracker's behavior. For example, a bias in the tracking error distribution may reveal a bias in the steering control for which the tracker must compensate.

APPLICATION TO PURE PURSUIT AND ZHANG TRACKERS

Our objective is to create a tracker that has a minimal amount of oscillation in the path, and minimizes overall tracking error. To achieve this goal we need a tracker that closely follows

curved and straight paths and has a fast settling time. When following a curved path, the tracker must closely follow the path so that the tracking error is not large when exiting the curved section. The tracker needs a fast settling time so after exiting a curved section the robot quickly settles onto the new path segment. To avoid visually obvious path oscillations we desire to keep the oscillation amplitude below 5 cm.

The three tests outlined above were used to compare the performance of a pure pursuit tracker and a tracker introduced in Zhang (1997). We wish to find which tracker has the fastest settling times, minimum tracking error vs. curvature and smallest tracking error when following straight lines. For these tests, we used a robot with an inverted tricycle design with 3 wheel drive and rear steering wheel. In the tests the localization system used dead-reckoning. The front wheels were equipped with encoders to measure distance traveled, and a fiber optic gyroscope was used to give heading information. These tests are meant to measure the performance of the tracking algorithms. To do so as accurately as possible all extraneous sources of error must be removed or reduced. Over short distances, the dead-reckoning system gave us the cleanest positioning information. If we had used a GPS system, the results of these tests could have been influenced by the GPS's errors.

Pure Pursuit and Zhang Trackers

In pure pursuit (fig 2), the algorithm finds a point along the path that is a look ahead distance away from the control point. The steering angle is proportional to the y coordinate of the look ahead point translated to the robot's reference frame.

$$\gamma = K_p * (-e_x * \sin(\theta) + e_y * \cos(\theta))$$

Where γ is the steering angle, θ is the robot's heading, $e_x = x_f - x_g$ and $e_y = y_f - y_g$

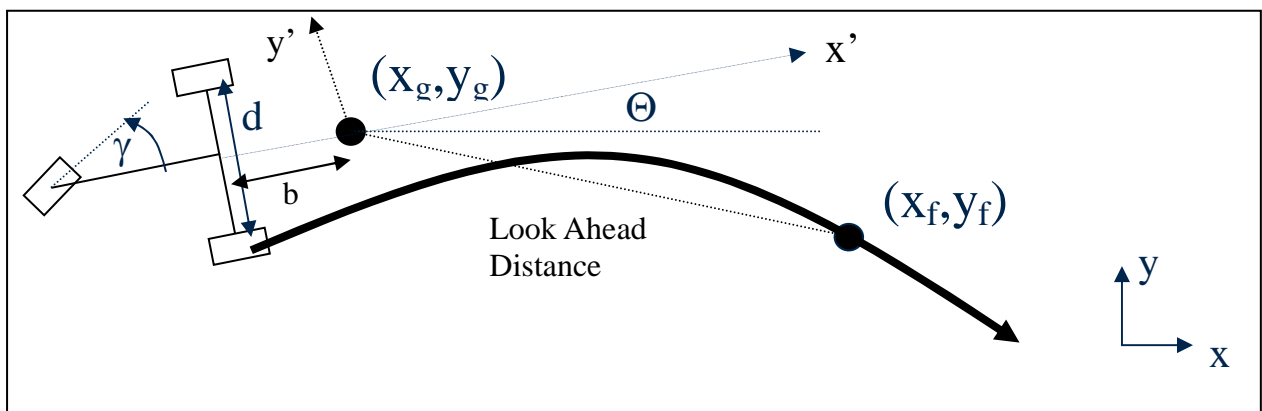


Figure 2: Pure pursuit finds a point (x_f, y_f) that is a look ahead distance away from the control point (x_g, y_g) . The steering command γ is proportional to the look ahead point's y' coordinate in the robot's frame.

The Zhang tracker was initially designed for use on a differentially steered robot. We could not control the angular velocity of the robot's front wheels directly, but we could indirectly by setting the steering angle. The control input of the Zhang tracker is defined by the following set of equations (Zhang et al. 1997)

$$\mathbf{u}(t) = \begin{bmatrix} \omega_l \\ \omega_r \end{bmatrix} = \mathbf{G}_p^{-1} [\dot{\mathbf{x}}_f + \mathbf{K}_p \mathbf{e}(t)]$$

$$\mathbf{G}_p = \begin{bmatrix} 2b \sin(\Theta) + d \cos(\Theta) & -2b \sin(\Theta) + d \cos(\Theta) \\ -2b \cos(\Theta) + d \sin(\Theta) & 2b \cos(\Theta) + d \sin(\Theta) \end{bmatrix}$$

The vector \mathbf{u} is the control input, ω_l and ω_r are the left and right wheel angular velocities, \mathbf{K}_p is a diagonal proportional gain matrix, b is the distance from the wheel axis to the control point and d

is the wheel base length. The x_f is the point on the path the tracker is following, and $e = [x_f - x_g, y_f - y_g]^T$ is the following error vector from the robot's control point to the following point.

Because we cannot control the wheel velocities directly, we derived an equation relating the wheel angular velocities to a steering angle (Kiriy 2001) (Fig 3).

$$\gamma = \tan^{-1} \left(\frac{\omega_l - \omega_r * \frac{2SA}{d}}{\omega_l + \omega_r} \right) \quad (1)$$

In this equation, ω_l , ω_r and ω_s are the left and right wheel angular velocities.

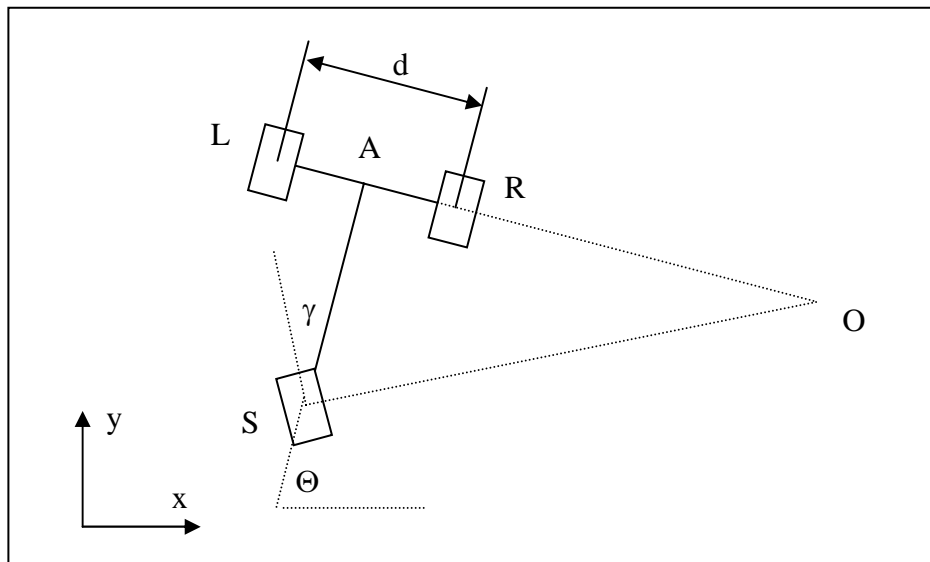


Figure 3: The kinematics of the robot are used to determine the steering angle as a function of the left and right wheel angular velocities.

Both of these trackers were slightly modified during these experiments. The steering control had a 0.25 second delay between when a steering command is issued and a response from the steering wheel. This delay is not significant when moving at slow speeds (<1 m/s), but becomes important at higher speeds. Both trackers compensate for this delay by using a kinematic model to calculate where the robot will be in 0.25 seconds. The tracking algorithms use this position as their input.

There were other modifications to the Zhang tracker. The output of the Zhang tracker is meant to control both the robot's forward and angular velocity. However, the robot's speed control is poor. A commanded change in speed starts the robot's speed oscillating, and it takes several seconds for these oscillations to stop. For this reason, we fix the robot's forward speed, and only use the Zhang tracker's output to command a steering angle. Because we fix the robot's speed, we also had to give the Zhang tracker a look-ahead point similar to the pure pursuit algorithm.

Results

Discontinuous Path

The discontinuous path test (fig 1a) is used to measure the trackers' stability, settling time and oscillation frequency at different speeds. We wish to maximize stability and minimize settling time and oscillation frequency. The robot's target speed for straight sections is 1.5 m/s so these tests were run at 1.5 m/s and 2.0 m/s. The results of the 1.5 m/s test are shown in figure 4. The Zhang tracker shows little oscillation in its response, and the pure pursuit shows a slight overshoot and subsequent oscillations. In both cases the system is acting like a damped oscillator. Due to the discontinuous nature of this path, we did not use the tracking error vs. time to measure the settling time and oscillation frequency. Instead, we used the difference in the following point's and robot's y coordinate as the new 'tracking error'. Figure 5 shows a plot of this

‘tracking error’ as a function of time. The solid line shows the fitted function $y = y_0 \exp(-t/\sigma) \cos(\omega t)$ where t is the time since the discontinuity, σ is the exponential decay constant and ω is the oscillation frequency. Table 1 shows σ and ω for the 1.5 m/s and 2.0 m/s cases. At 1.5 m/s, the Zhang tracker and pure pursuit have the same decay time, but the Zhang tracker has a lower oscillation frequency. At 2.0 m/s, the pure pursuit tracker was not able to recover from the discontinuity, and it became unstable. The Zhang tracker’s decay time and oscillation frequency increased, but it was still stable. In both tests, the Zhang tracker has better settling times and oscillation frequencies and is more stable.

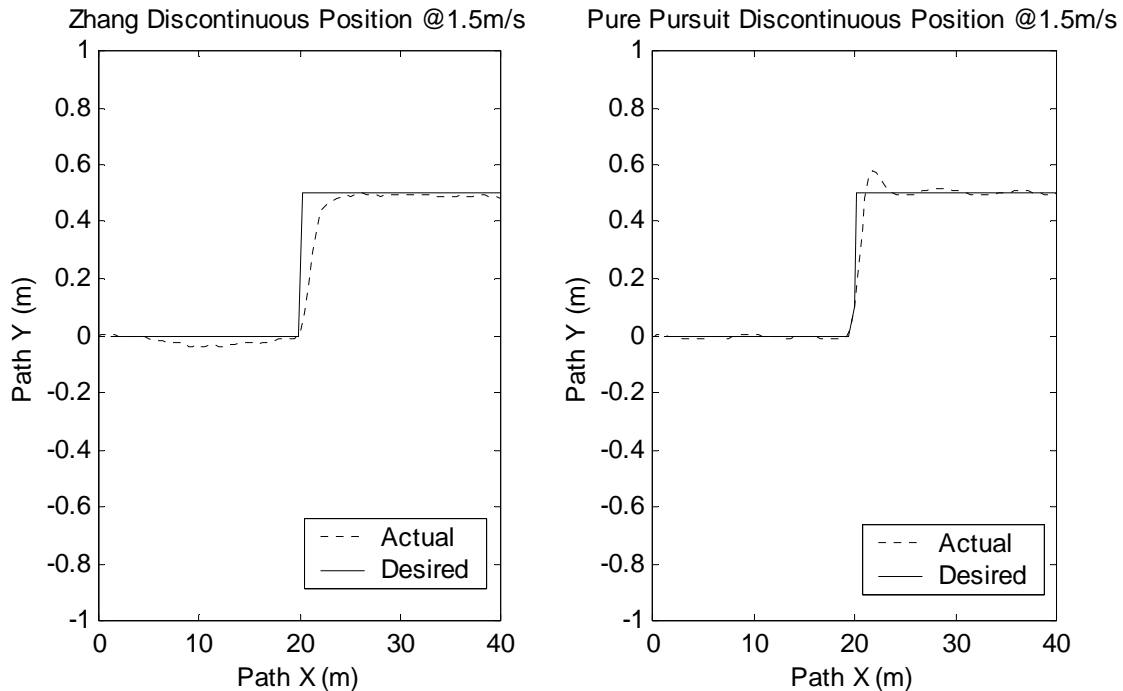


Figure 4: This figure shows the desired (solid) and actual (dashed) paths taken in the 0.5 m step path tests run at 1.5 m/s. The Zhang results are shown on the left and pure pursuit on the right.

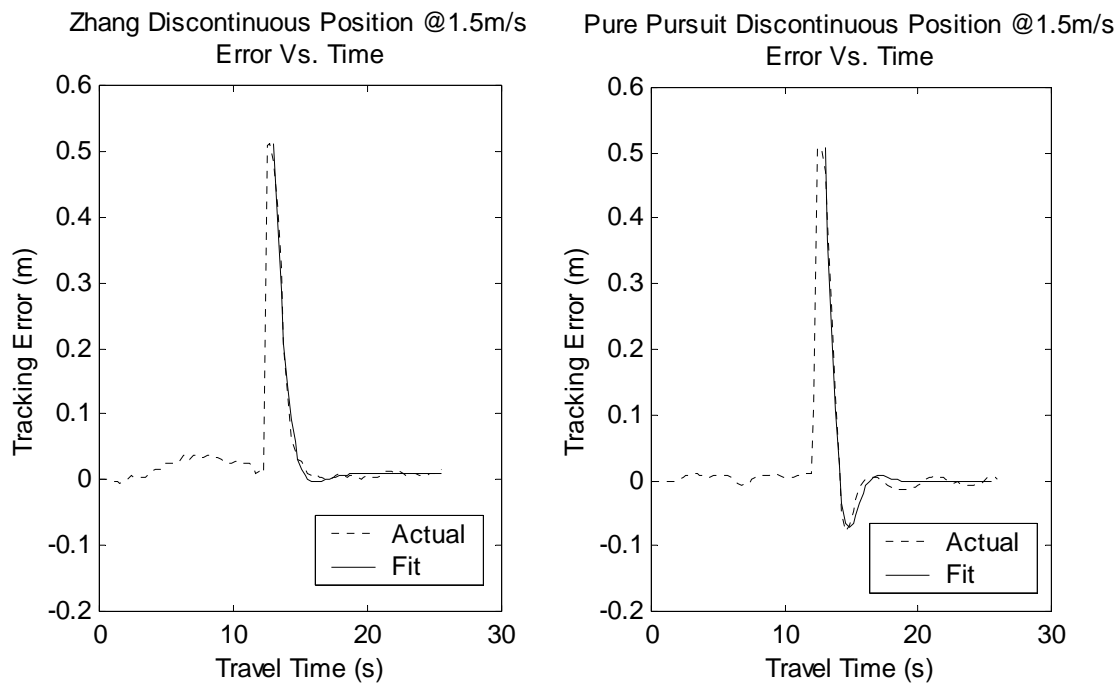


Figure 5: This figure shows the tracking error and a fitted function $y = y_0 \exp(-t/\sigma) \cos(\omega t) + y_1$ for the discontinuous path test at 1.5 m/s. Once again, the Zhang results are on the left and pure pursuit results are on the right.

Speed (m/s)	Tracker	σ (seconds)	ω (seconds ⁻¹)
1.5 m/s	Zhang	1.01	0.71
1.5 m/s	Pure Pursuit	1.03	1.40
2.0 m/s	Zhang	1.65	2.30
2.0 m/s	Pure Pursuit	Unstable	Unstable

Table 1: This table shows the decay times and oscillation frequencies for the step discontinuity path tests for the Zhang and pure pursuit trackers. The Zhang tracker is more stable, has a shorter settling time and smaller oscillation frequency.

Curvature Test

The curvature test (fig 1b) is designed to measure the tracker's error as a function of path curvature, and represents how well the tracker performs when following curved paths. We wish to minimize the tracking error vs. curvature to ensure the robot is not too far from the path when exiting a turn. These tests were run at the robot's target turning speed 1.0 m/s. The tests used circular paths with 5 m, 2.5 m and 1.7 m radius, and a straight line. The 1.7 m circle was the tightest circle the robot could follow without saturating the steering.

Figure 6 shows the tracking error as a function of time for the 1.7 m circle. As the robot goes around the circle, the tracking error settles to a steady state value. It is the mean of this steady state tracking error that we measured as a function of path curvature in figure 7. The error bars are +/- one standard deviation of the data, and represent the amount of oscillation. Both trackers follow the paths with approximately the same amount of tracking error. However, the Zhang tracker oscillates more in higher curvature turns. Although the Zhang tracker does oscillate twice as much as pure pursuit at high curvatures, the magnitude of the Zhang tracker's oscillation is only +/- 3 cm vs. +/- 1 cm for the pure pursuit, a relatively small oscillation.

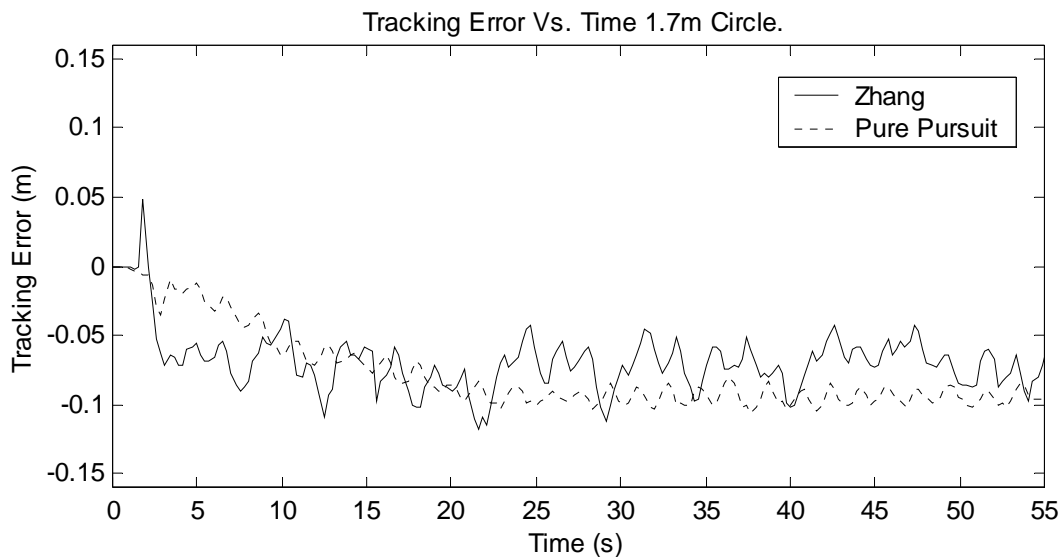


Figure 6: Plot of tracking error vs. time for the 1.7 m circle path. The tracking error settles to an average tracking error as the robot travels around the circle.

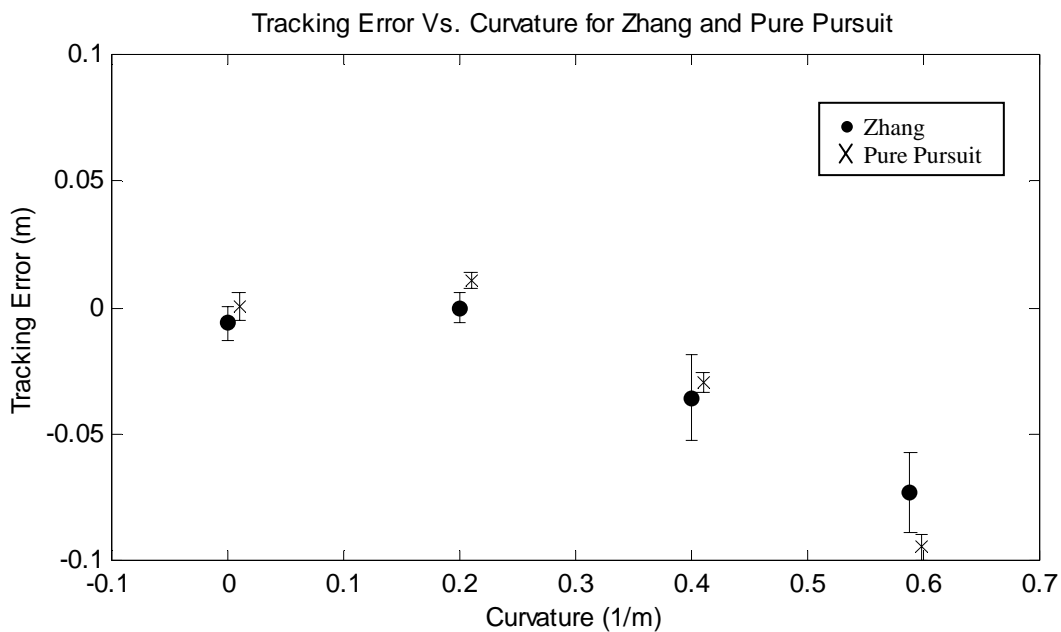


Figure 7: Mean tracking error vs. curvature for the Zhang and pure pursuit trackers. The data points represent the mean steady state tracking error, and the error bars represent 1 standard deviation.

Real World Test

The real world test's purpose is to measure the trackers' performance under pseudo-real-world conditions (fig 1c). Because we are most concerned about the trackers' straight-line performance that is where we concentrated our analysis. We wanted to measure the tracking error along the path's straight sections, but we did not want to include the path segments that follow right after a turn. Immediately after a turn, the robot was slightly off the path, and this offset induced oscillations that slowly faded. We measure the tracking error along straight sections beginning 2 m after a turn. We wish to determine which tracker has the smallest straight line tracking error. Figure 8 shows a histogram of the tracking error for the path's straight segments. The Zhang histogram shows a tall narrow peak while the pure pursuit's histogram is shorter and wider. The Zhang tracking error has a 0.0094 m standard deviation and spans the range [-.033 m, 0.020 m]. The pure pursuit has a 0.014 m standard deviation and spans [-0.043 m, 0.028 m]. The Zhang tracker's narrower peak means there is less oscillation in the straight paths. The difference in standard deviations between the two trackers is only 0.5 cm so the Zhang tracker only has a slight advantage over pure pursuit when following straight lines. The tracking error in both plots has a slight bias. The mean value for the Zhang tracker is -.006 m and -.004 m for pure pursuit. This bias is noticeable on the plots, but insignificant when visually inspecting the overall path. The trackers have very similar performance characteristics with the Zhang tracker holding a small advantage.

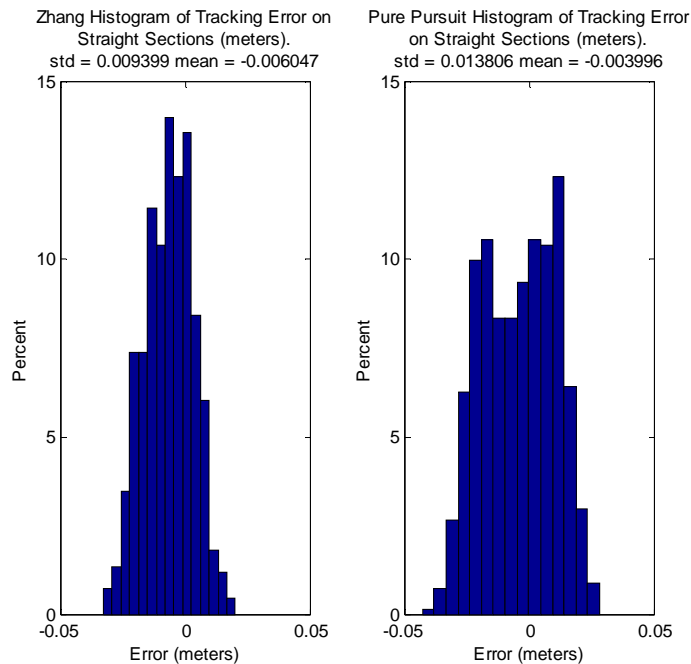


Figure 8: This figure shows histograms of the tracking error for the real world path's straight sections

CONCLUSION

High accuracy path tracking is very important in some applications. Oscillations in the paths must be kept below 5 cm to avoid visually obvious errors. A tracker must also closely follow curved paths and have a fast settling time to minimize the time between leaving a turn and settling onto a straight path.

To achieve these goals, we developed a series of three tests designed to measure a tracker's performance. The tests measured the tracker's stability, settling time, oscillation frequency and tracking error using three different path types. The stability, settling time and oscillation frequency were measured using a path with a step discontinuity. The tracking error on curves was measured using circular paths of different radii, and the tracking error on straight sections was measured using a 'real world' path made up of straight and curved sections.

Overall, the Zhang and pure pursuit trackers have similar performance characteristics, and both perform within our specifications. However, the Zhang tracker has a slight advantage due to a faster settling time, greater stability and smaller tracking error over straight paths.

For future work, both path trackers can use some improvements. Because the two trackers' performances are similar, it would probably be simple to adjust the pure pursuit algorithm to meet the Zhang tracker's performance. For example, proper gain scheduling dependent upon tracking speed and path curvature could improve the pure pursuit's performance in the step path test. The Zhang tracker's largest weakness, oscillations on curved paths, could also be improved by similar gain scheduling.

Acknowledgments

The authors would like to thank Sanjiv Singh for his support in this work. We would also like to thank Martin Buehler and Evgeni Kiriya at McGill University for their work adapting the Zhang tracker to our application.

REFERENCES

1. Baltes, J. 2000. A Benchmark Suite for Mobile Robots. *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems*: 1101-1106.
2. Garbutt, J., T. Vinay, and S. John. 2000. Evaluation of Controller Schemes for a WMR by Computer Simulation. *Australasian Matlab Users Conference*:
3. Kiriy, E., M. Buehler. 2001. Automated Turf Management Progress Report (Q1, 2001). *Internal Document*
4. Mouri, H., and H. Furusho. 1997. Automatic Path Tracking Using Linear Quadratic Control Theory. *IEEE Conference on Intelligent Transportation Systems*: 948-953.
5. Petrov, P. 1991. Robust Trajectory Tracking Algorithms for a Wheeled Mobile Robot. *IEEE International Conference on Systems Engineering*: 1071-1074.
6. Sarkar, N., X. Yun, and V. Kumar. 1994. Control of Mechanical Systems with Rolling Contacts: Applications to Mobile Robots. *International Journal of Robotics Research*: 55-69
7. Shin, D. H., S. Singh, and W. Shi. 1991. A Partitioned Control Scheme for Mobile Robot Path Tracking. *IEEE International Conference on Systems Engineering*: 338-342.
8. Weiguo, W., and Y. Wang. 1999. Backstepping Design for Path Tracking of Mobile Robots. *International Conference on Intelligent Robots and Systems*: 1822-1827.
9. Zhang, Y.L., S. A. Velinsky, and X. Feng. 1997. On the Tracking Control of Differentially Steered Wheeled Mobile Robots. *Journal of Dynamic Systems, Measurement, and Control*: 455-461.