Documentation on NORTHSTAR

Jeremy Ma
PhD Candidate
California Institute of Technology
jerma@caltech.edu
June 7th, 2006

1 Introduction

One of the most difficult aspects of coordinated control of mobile agent(s) is obtaining consistently accurate position estimates of the robot. Many methods currently exist that provide a reasonable estimate of the robot pose (e.g. wheel odometry, GPS, scan-matching methods), yet none that stand alone reliably, consistently, and accurately at each moment in time. This documentation addresses the use of an indoor GPS system known as "Northstar" which has been developed by Evolution Robotics. Discussion of the configuration, accuracy, and modifications of the Northstar system into our current testbed of mobile agents will be presented as well as how Northstar has been fused with our existing pose measurements through a Kalman Filter to yield the optimal position estimate. Data results from two tests illustrating the accuracy of Northstar and the Kalman Filtered pose estimates are also provided along with some discussion and conclusion towards future work implementing new localization methods for more accurate pose estimates.

2 Background

The Northstar Detector-Projector Bundle Kit comes with a detector and a projector. The detector itself is an optical sensor about the size of a matchbook that detects infrared light. The projector is composed of two platforms, with each platform containing four LED lights that blink at particular frequencies tuned to be registered accurately by the detector [0] . The two platforms are required to establish differential GPS. The output of the Northstar system is a string of six values: an x, y, and instensity value for each of the two platforms. With these six fields, the robot position in an x-y plane and its orientation can be deduced.

The Northstar system can be used in two different configurations: the Asset Tracking Configuration and the Navigation Configuration. The Asset Tracking Configuration consists of a detector fixed to the ceiling and a single projector fixed to each mobile agent. The Navigation Configuration consists of two spots projected onto a flat ceiling with a single detector fixed to each mobile agent. Evolution Robotics originally designed the system with the Navigation Configuration in mind, which is optimal for high, flat ceilings with non-reflective surfaces. However, due to some complications regarding the geometry of our laboratory and the type of OS used for each of our mobile agents[1], we were forced to use the Asset Tracking Configuration.

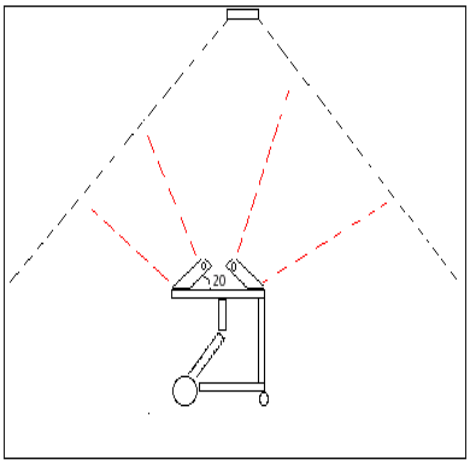


Figure 1: Asset Tracking Configuration

In the Asset Tracking Configuration, the detector will detect all projectors in the room blinking at user-set frequencies. However, since each projector platform has four LED lights, the detector has a tendency to saturate if a projector platform is aligned directly below or facing it. A fix we implemented to get around this was to have each projector platform tilted at an angle (current angle setting on each robot is 20 degrees from the horizontal). This prevents the robot from flooding the detector with light intensity when the robot is sitting directly below the detector. Consequently, it is not until two meters

[0] Note that when referring to the projector, the particular platform of interest should be made clear by the context.

[1] Currently, our testbed consists of six ER1 robots: two wheeled non-holonomic vehicles purchased from Evolution Robotics

out that the projector becomes directly aligned with the detector, which is acceptable since at that distance, the light intensity is not strong enough to saturate the detector as shown in Fig.1.

Complications have arisen when two or three robots are within the field of view and each robot is directly looking at the detector. In such cases, the detector will saturate and small movements in the robots' pose will not be detected. Thus it becomes a balancing act of keeping robots away from directly facing the detector to avoid saturation while at the same time keeping them relatively orientated in the direction of the detector in order to keep a strong detection of signals.

When using the Northstar system in the asset tracking configuration, all projector locations (whether it be one projector set for one robot or multiple sets for multiple robots) are seen by the single detector fixed to the ceiling. We had the detector connected via a serial cable to a desktop in the back of the room which was connected to a wireless network shared by all the mobile robots. We then modified the Northstar source code to broadcast every measurement update for every detected robot over the network to each robot using a communication architecture known as Spread[2]. Mobile agent pose updates were measured to occur every 0.2804 seconds.

[2]www.Spread.org

3 The Calibration Constant

In calibrating the Northstar system we took a relatively ad hoc approach. Several factors had to be considered that could ultimately affect the accuracy of the robots' pose:

- Northstar outputs data in non-dimensional units that need to be calibrated into SI units. Assuming the x and y values of a robot's pose to be independent, then separate calibration constants must be determined for the x and y directions.
- Northstar reports data measurements relative to the location of the origin of the detector, which is difficult to perfectly align with any coordinate frame that may already exist on the lab-room floor.
- In the Asset Tracking Configuration, there is a field of view for which the detector is limited to detect signals. It is a function of the height at which the detector is fixed at, relative to the projector elevation and needs to be determined.
- There are two platforms on each robot, with each platform tilted away at an angle.

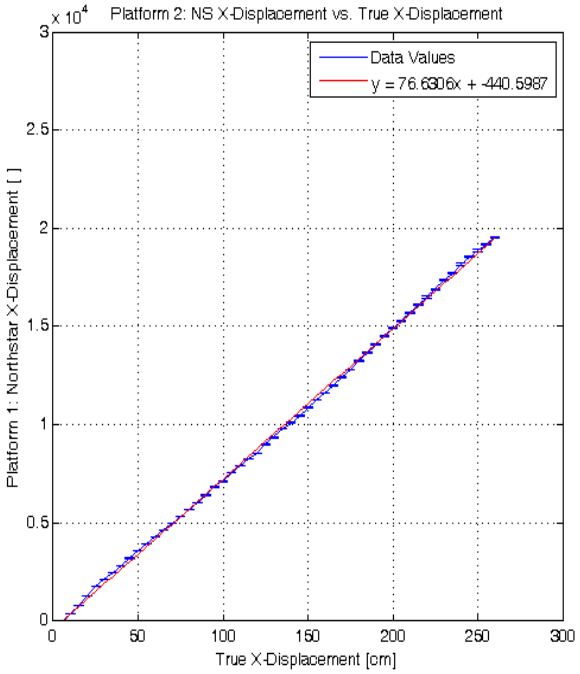
As a first step, we defined an $x - axis$ and a $y - axis$ on the floor of the laboratory with the origin directly beneath the detector. We then placed a single ER1 robot aligned with the positive $x - axis$ and took data measurements in intervals of 5cm, moving along the positive $x - axis$[3]. At each displacement interval, the robot was left stationary and 100 data points were collected from Northstar for each platform. From each set of 100 data points, we extracted means, standard deviations, and variances. We did the same approach for the $y - axis$, moving positively in the y-direction, recording data for each platform [4]. We then plotted the mean and σ-uncertainty[5] at each 5cm increment for both directions (first for x, then for y) against recorded Northstar data. We expected a linear relationship and the slope of which would be the calibration constants in the x-direction and the y-direction.

As can be seen from Fig.2, Fig.3, Fig.4, and Fig.5, the platform that was always facing the detector had the strongest sense of localization from Northstar even beyond the full length of the lab floor. It also had the lowest σ-uncertainty in x and in y. The platform which was not facing the detector had a much lower sense of localiztion from Northstar and had a larger σ-uncertainty than the other platform. It also lost its sense of localization somewhere between 2m - 2.5m. Thus, we decided to go with the calibration constant determined from the platform facing the detector for both the x-direction and the y-direction. From the data collected, it was clear that Northstar would only be as good as its weakest platform measurement which indicated that the field of view for the detector was a circle of radius of 2m, centered at the origin. Once the calibration constant was determined, it was a simple translational correction to account for the offset between the presumed origin drawn on the lab floor and the actual origin used within Northstar.

[3] Note here that because the robot was aligned directly with the $x - axis$, there was always one projector platform facing the detector and one not facing the detector. This biases the measurements obtained, making the uncertainty of each platform a function of orientation

[4] Also note that due to the limited size of the labroom floor, we were only able to measure up to roughly 250cm in x and 300cm in y

[5] Note that σ-uncertainty and standard deviation will be used interchangeably throughout the course of this document in the discussion and in the figures



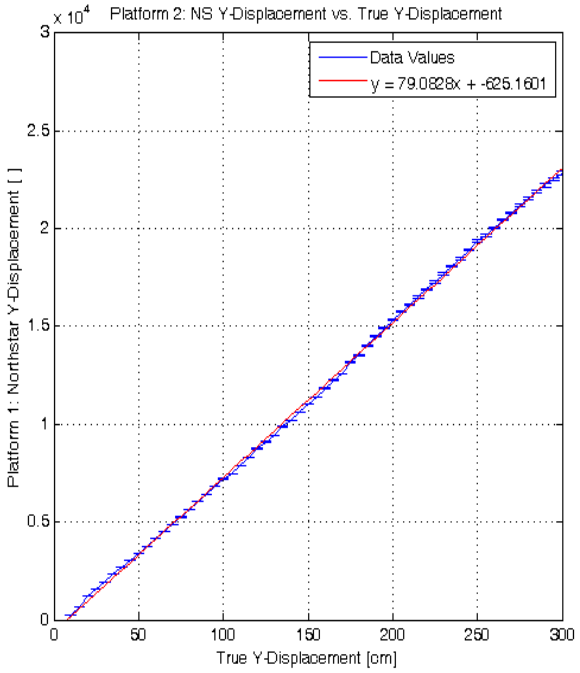
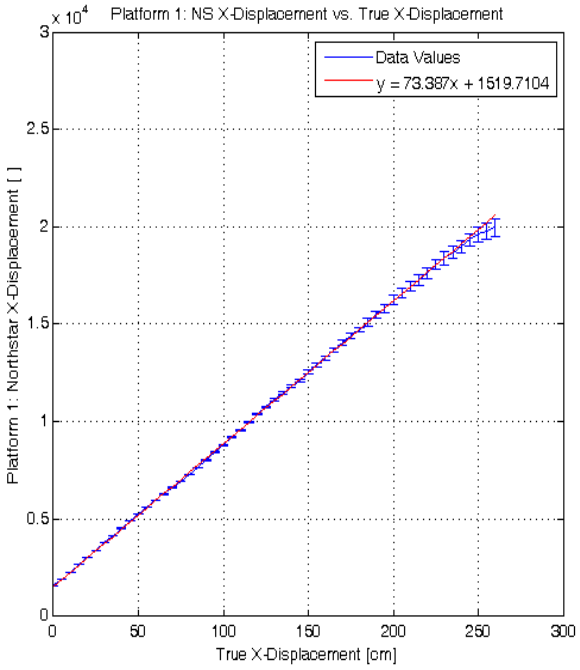


Figure 2: Facing platform results for Calibration Constant



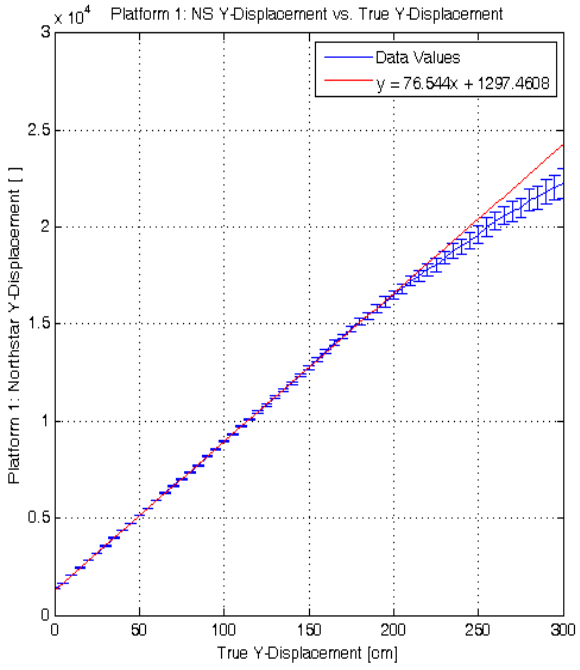
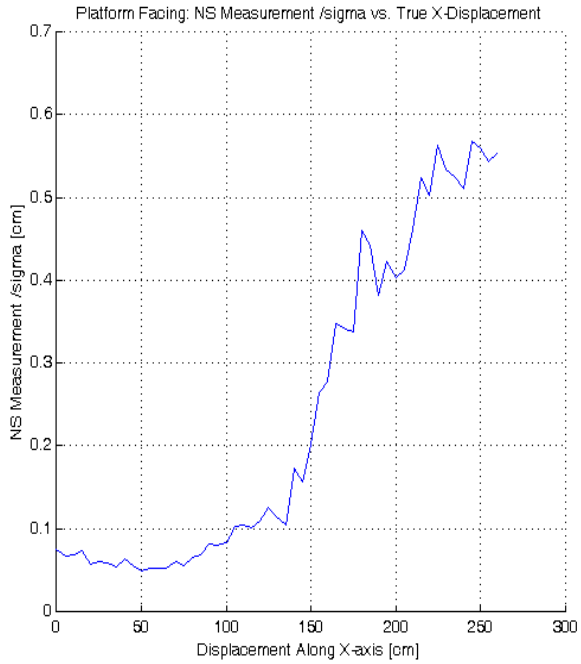


Figure 3: Not-facing platform result for Calibration Constant



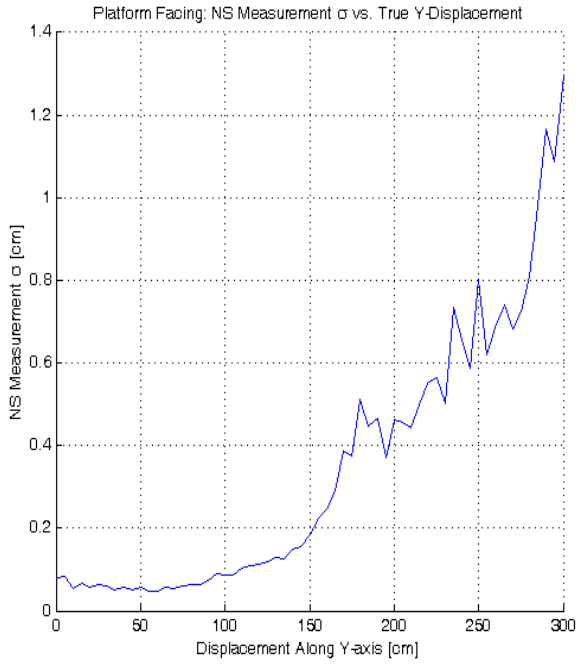
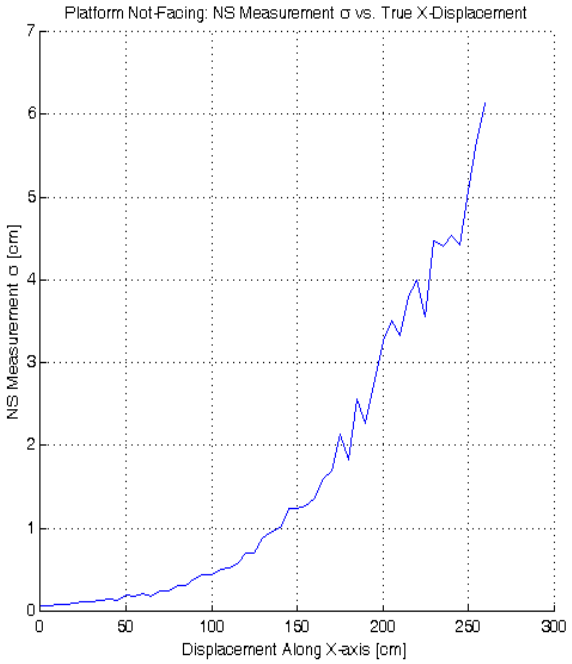


Figure 4: σ vs. Distance[cm] for Facing Platform



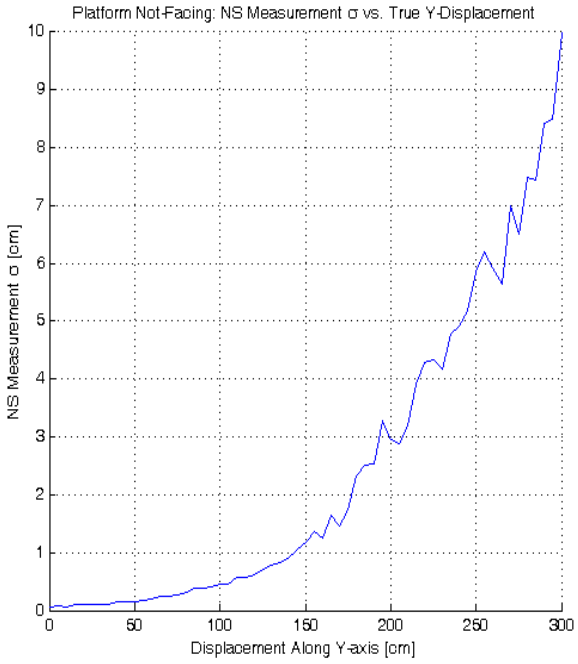


Figure 5: σ vs. Distance[cm] for Not-Facing Platform

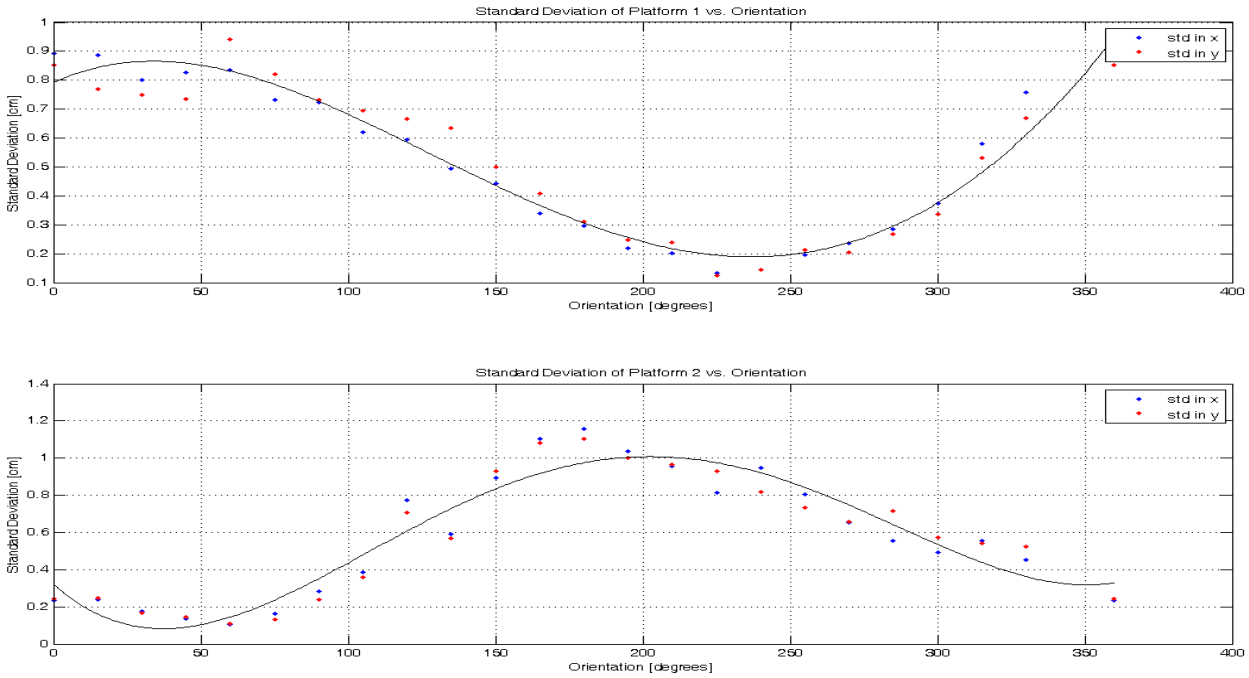


Figure 6: Standard Deviation vs.Orientation

For completeness of our analysis of the Northstar unit, we did one more test to quantify the uncertainty in robot pose as a function of orientation. We placed a robot equipped with a Northstar projector unit a fixed distance of (1m, 1m) from the detector. We then rotated the robot every 15 degrees and collected 100 data points on each platform at each orientation. We then plotted the standard deviation of each data set as a function of robot orientation (see Fig.6). The plotted results indicated that the two platforms had σ-uncertainty profiles that were sinusoidal in nature and out of phase with one another. This was expected since the smallest uncertainty of one platform was at an orientation facing the detector, consequently leaving the other platform oppositely-facing the detector and out of phase.

4 The Kalman Filter

Each ER1 robot equipped with a Northstar Projector Kit has access to an internal pose measurment from the wheel odometry (i.e. by counting the number of wheel rotations on each wheel, the robot will have a sense of localization). However, that internal measurement grows unbounded in error unless fused with some other external measurement. Thus, with Northstar, we are provided with that external measurement and can fuse that knowledge with the odometry estimate using a Kalman filter.

At this point, something should be said about how the two platform pose values were combined to yield one robot pose estimate. Since the projector consisted of two platforms, a weighted average of both platforms was used to produce a single estimate of the robot pose in x and y. To determine the appropriate weights, we simply used the intensity values for each platform (scaled appropriately to yield weights between 0 and 1, where the sum of both weights was unity).

The heading estimate was a rather crude estimate. It used a simple arctangent calculation that did not use any weights. To account for the uncertainty in this calculation, the θ element of the measurement covariance matrix, R, in the Kalman Filter ended up being used as a tuning parameter which we set to be much larger than the x and y R-covariance values.
The Kalman Filter predictor equations are shown below for reference:

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1} \tag{1}$$

$$P_k^- = AP_{k-1}A^T + Q \tag{2}$$

And the Kalman Filter update equations are shown below for reference:

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \tag{3}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^- \tag{4}$$

$$P_k = (I - K_k H)P_k^- \tag{5}$$

The device server we use on our mobile agents is a free-source package known as Player. Player acts as an interface between hardware and developed source codes with an extensive library of drivers for a large selection of sensors and actuators. Within Player, the wheel odometry output for the ER1 robots has a model of the dynamics built within the Player drivers. Consequently, we used the wheel-odometry output as the value of the prediction of our state, $\hat{x}_k^-$, in Eq.1 and treated the process noise covariance matrix, Q, as a tuning parameter. The measurement covariance matrix, R, could fortunately be determined using the Northstar intensity values since from the analysis of collected data sets, the intensities reported for each platform from the detector were inversely proportional to the standard deviation of the mean in x and in y via some proportionality constant (roughly a value of 90).

As can be seen in Fig.7, a scaled inverse of the intensity values matches quite closely to the largest σ-uncertainty in either the x or y direction for each platform. We used this scaled inverse of the intensity value to represent the x and y variance terms in our measurement noise covariance matrix, R, assuming zero cross-correlation between x and y. However, from the analysis shown in Fig.6, it is obvious that one platform would have a lower measurement covariance than the other if it had a better tilt towards the detector. Thus we decided that we would use the larger measurement covariance determined from

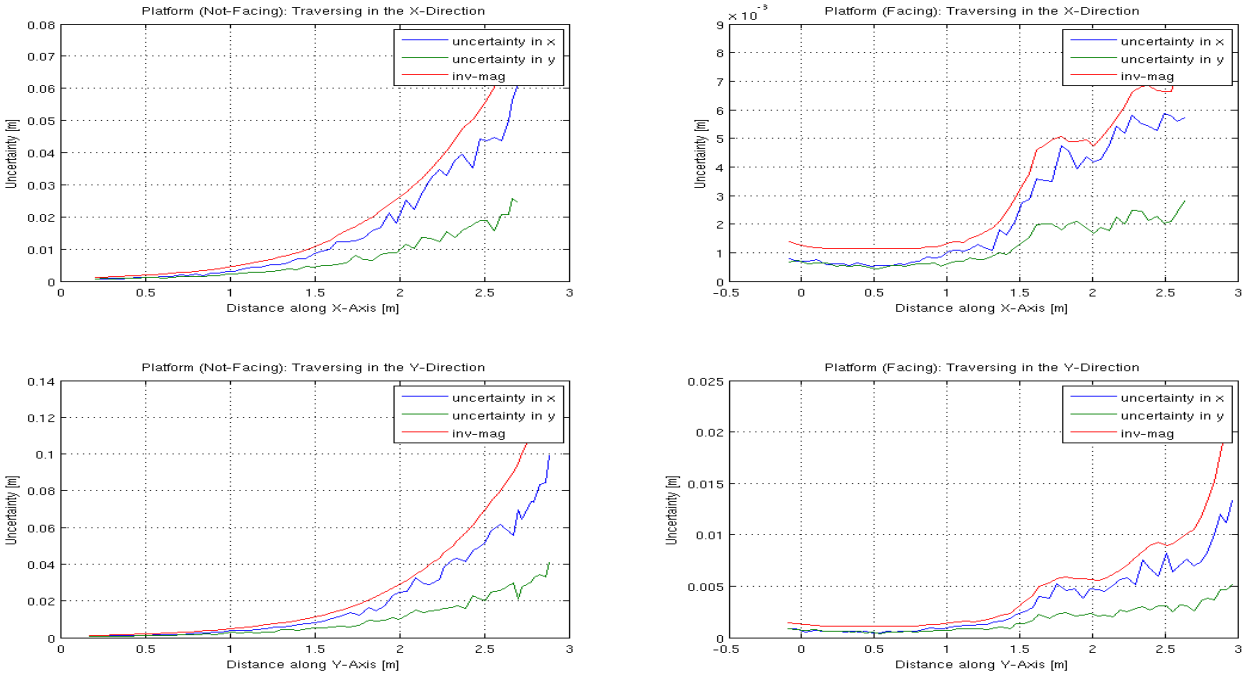


Figure 7: Plot of standard deviation vs. distance

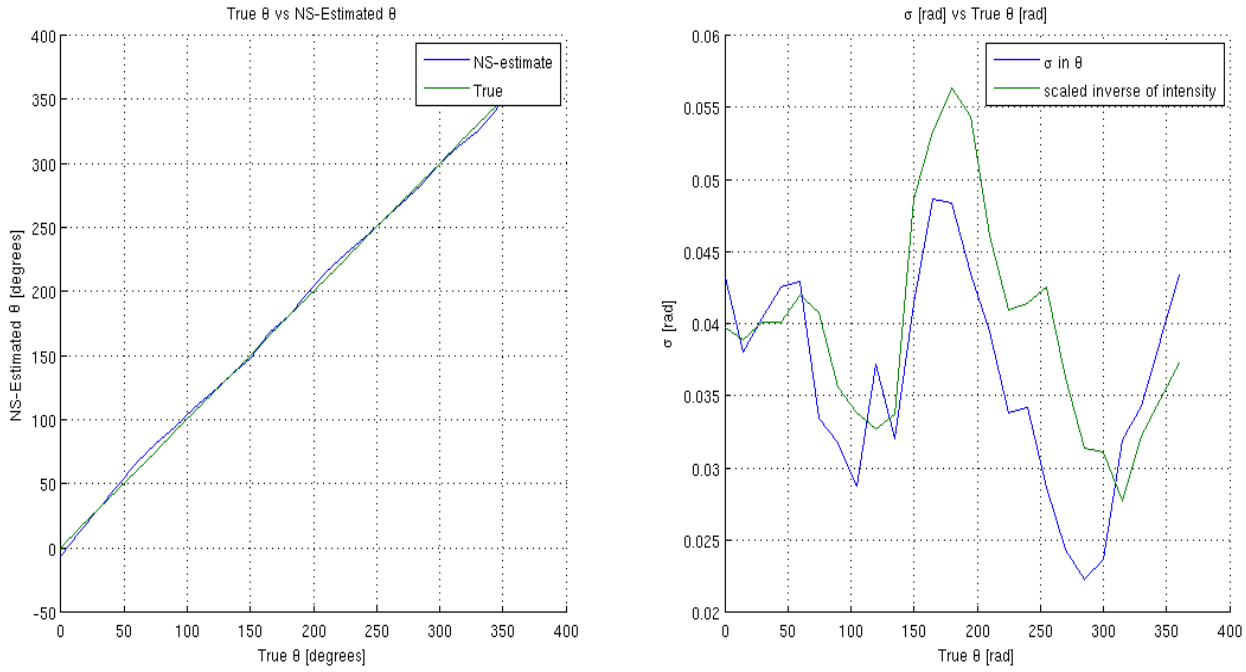


Figure 8: Plot of σ-uncertainty in θ from NS vs. True-θ

either platform as the measurement covariance values in both x and in y in the construction of the R matrix.

Concerning the θ variance terms in the measurement covariance matrix, R, we could not make the assumption that θ was independent of x and y, since θ would be determined by using both the x and y values of each platform in an arc-tangent calculation. Thus, as a first approach to quantifying these variance terms, we took the same Northstar data from the test used to generate Fig.6 and extracted a θ-estimate to plot against the known orientation values of the robot, via the following equation (where x_1, y_1, x_2, y_2 represent the x and y values of the first and second platform, respectively):

$$\theta = tan^{-1}(\frac{y_1 - y_2}{x_1 - x_2}) \tag{6}$$

We then used the following error-propagation equation to determine the σ-uncertainty in the calculated θ measurement, where the σ-uncertainty in x and in y are already known:

$$\sigma_\theta = \sqrt{(\frac{\partial\theta}{\partial y_1}\sigma_{y_1})^2 + (\frac{\partial\theta}{\partial y_2}\sigma_{y_2})^2 + (\frac{\partial\theta}{\partial x_1}\sigma_{x_1})^2 + (\frac{\partial\theta}{\partial x_2}\sigma_{x_2})^2} \tag{7}$$

Plotting these uncertainty values against known heading values, we arrived at Fig.8. Note that in the figure, the right-sided plot shows an additional curve, which is a scaled inverse[6] of the lower intensity of the two platforms. Now since the scaled inverse of the intensity value was very similar in size and shape to the σ-uncertainty plot associated with Northstar's measurement of heading, we made a very crude estimation that the θ-variance term in the R measurement covariance matrix could be approximated by this scaled inverse of the magnitude (a proportionality constant of 100 was used here). Furthermore, to keep the calculations simple, we also made another assumption that the cross-correlation terms involving θ could be neglected as small (esentially zero) in comparison to the larger diagonal elements that would contribute the most to the Kalman Filter.

Lastly, to account for the field of view of the detector (which we determined to be a circle of radius 2m), we included a conditional statement within the Kalman Filter that would set the covariance of the measurement, R, to infinity if the robot were to ever traverse outside the field of view. Thus for distances outside the 2m radius circle, the robot relied solely on wheel odometry.

A copy of the source code implemented in our testbed is given in Appendix A for reference.

[6] a different proportionality constant than the one used for the x and y uncertainty values in the R covariance matrix

5 Data Analysis

We performed two trials to observe the accuracy of Northstar when fused with wheel odometry via a Kalman Filter. Both trials consisted of starting a single robot at the origin of the lab room and moving it to predetermined points along the lab room floor, in incrememental distances of roughly 25cm. In the first trial, the robot paused for 15 seconds, allowing enough time to record pose data from wheel odometry, Northstar measurements, and the Kalman filtered estimates of the robot pose. In the second trial, the robot was held stationary long enough to collect an additional measurement from an alternate localization method known as scan-matching (more on this later). Ground truth data was also collected at each position for both trials.

5.1 Trial 1

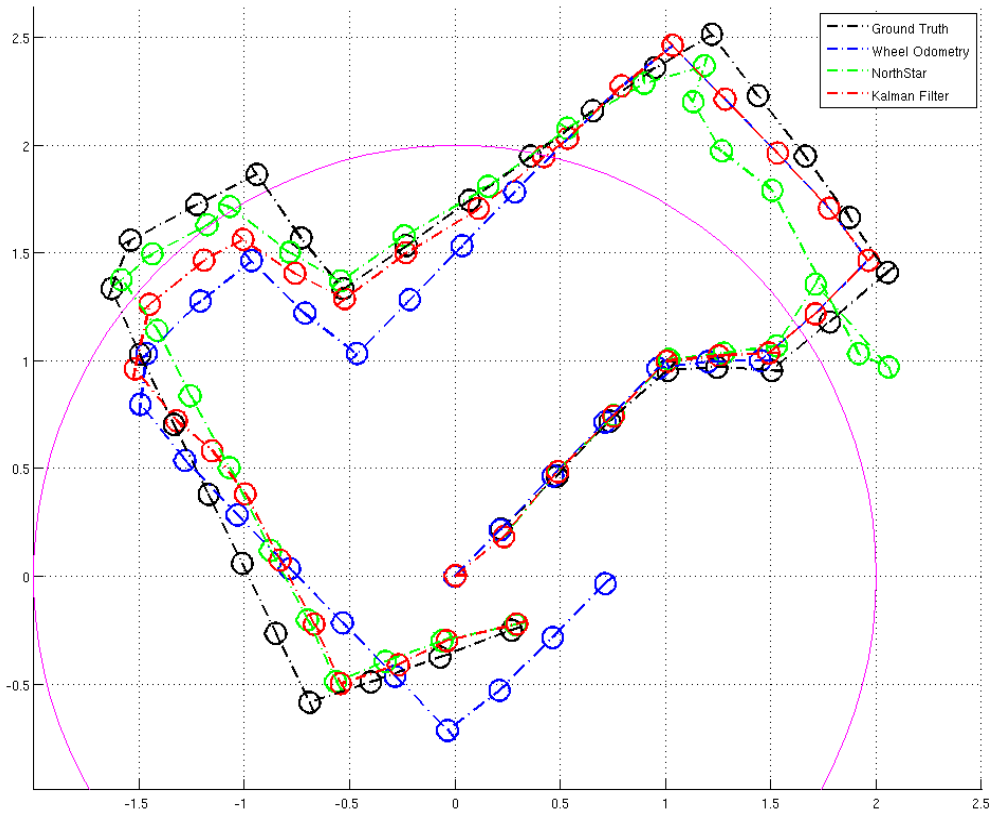


Figure 9: Northstar Comparison Test Results

As can be seen from Fig.9, the Northstar measurements started to get slightly skewed after about a 1.5m out and there was already some observed slipping in the odometry[7]. Once the robot had travelled past 2m, note that Northstar became fairly inaccurate and that the wheel odometry and the Kalman Filtered pose estimates were identical. This was expected since we had modified the Kalman filter to place infinite measurement covariance on Northstar measurements of x, y, and θ once the robot had travelled outside the field of view (a circle of radius 2m), and to rely solely on wheel odometry. However, once the robot came back into the field of view, we saw that the Northstar measurement became more reliable and wheel odometry became more inaccurate. Despite the errancy of the wheel odometry and the limited field of view of the Northstar detector, we saw that these factors did not significantly affect the Kalman filter estimates for the majority of pose estimates, with the Kalman estimate being relatively close to the ground truth.

The second time the ground truth data showed the robot to have left the field of view (in the second quadrant), we saw that the Kalman filtered data still determined the robot pose to be within the circle of radius 2m. This indicated that the measurement covariance of Northstar had not been set to infinity and the Kalman filter was still using Northstar data to fuse with odometry measurements despite that Northstar was unreliable. What was interesting though, was that once the robot re-entered the field of view, there was a bias to the Northstar measurements which lasted almost all the way back to the origin. However, despite this anomaly, the Kalman Filter was doing a good job of weighting the Northstar measurements over the wheel odometry for the majority of the run (i.e. we see the red circles almost always very close to the green circles).

We investigated the bias offset seen around the second quadrant of Fig.9 and determined two possible causes to the offset:

- A dry-erase board surrounded by several doors and cabinets with reflective windows may have been reflecting the infrared light beams from the projector to other locations around the room causing the detector to falsely identify the position of the robot. Preliminary tests using black felt cloth to cover these reflective surfaces did help to correct the offset but did not fix it completely.

- A large air conditioning unit that sits on the ceiling over the second quadrant may have blocked a majority of the infrared light beam coming from the one platform not facing the detector. Since the two platforms are weighted according to their intensity values, the one platform that can be sensed gives a strong bias that the robot center is shifted in that direction.

5.2 Trial 2

The purpose of this second trial was to verify the results found from the first trial and to observe the accuracy of a new localization method introduced onto our testbed, known as "scan-matching". Scan-matching is an algorithm often used to provide consistent accurate pose estimates first presented by Feng Lu and Evangelos Milios[8]. It takes two-dimensional laser range scans at two different locations

[7] The slipping was a result of the location of the predetermined points. They were chosen in such a way that would require the robot to issue large turnrates which is more inducive to slipping

[8] *Lu, Feng and Evangelos Milios.* Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans. *Journal of Intelligent and Robotic Systems, 1997.*

and estimates the optimal relative pose between the two scans that minimzes the least-square-error between each correlated point in the scan.

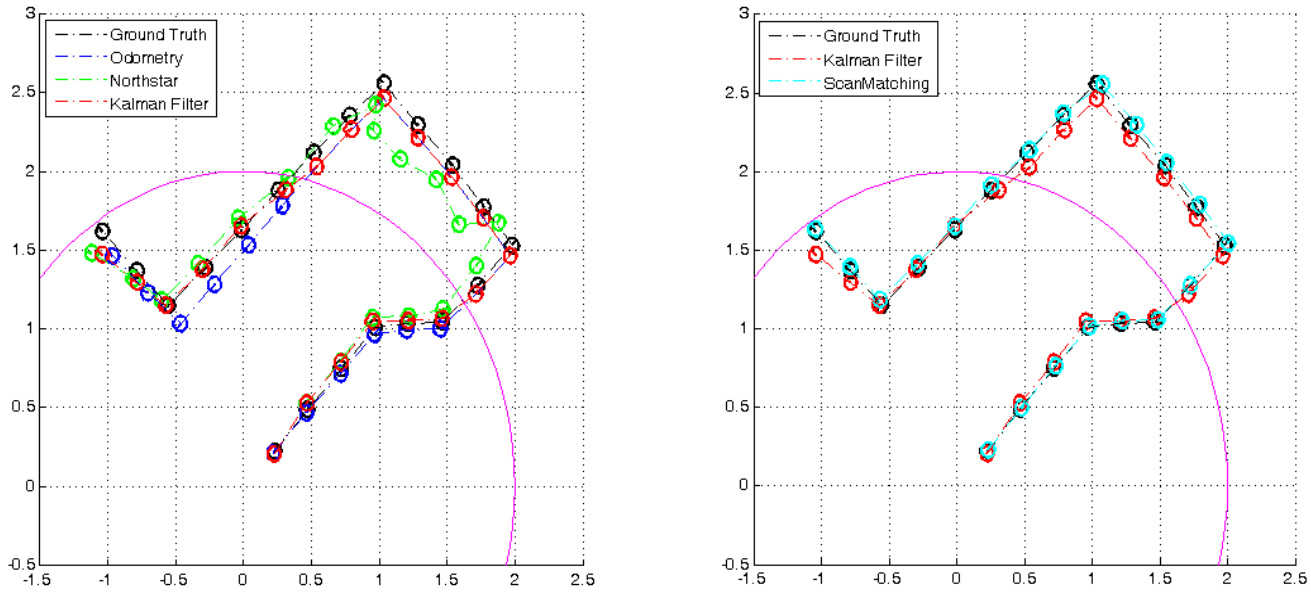


Figure 10: Northstar Comparison Test Results with Scan-matching

The first figure on the left of Fig.10 is for the most part identical to Trial 1 with fewer points, and corroborates the results and analysis done in the first trial. In comparison, the figure on the right illustrates the observed results of using scan-matching for localization. As can be seen in the figure, the raw scan-matching algorithm of Lu-Milios (which does not make use of any noise models, nor is it filtered with any other measurement data) does a remarkably good job of estimating the pose of the robot when compared to ground truth and outperforms the Kalman Filtered results (where only wheel odometry and Northstar are used in the filter).

6 Summary

From the extensive analysis done on Northstar and the results of Trial 1 and Trial 2, we observed a few key observations:

- Northstar is a system to be used in the Navigation configuration
- When used in the Asset Tracking Configuration, the Northstar Projector Kit is extremely sensitive to reflective surfaces and each reflective surface needs to be accounted for to reduce bias effects on the measured pose
- The Kalman Filter proved to provide a reasonable estimate of the robot pose in places where one sensor measurement may not have been as reliable as the other.

From the promising results of Trial 2, we intend to augment the current scan-matching algorithm with accurate noise models[9] and fuse the pose estimates obtained from this algorithm with other collected data into a larger Kalman Filter. The resulting pose estimates should be much more accurate then the results presented here.

One thing that we had not discussed much in this documentation was the accuracy of the Northstar measurement when more than one robot is equipped with a projector. From informal tests and analysis, we've observed that the more robots in view of the detector the greater the possibility of saturation and the lower the accuracy of pose measurements. There are some improvements that can be made to the accuracy of the Northstar measurement, yet we were unable to explore these improvements due to the geometry of our laboratory and the time involved. Certainly higher ceilings would reduce saturation effects and the possibility of making some omnidirectional configuration of the LED infrared sources would reduce bias effects associated with orientation. But from the analysis shown and the breadth of experiments we desire to carry out in our laboratory, it is evident that Northstar alone will not provide the necessary accuracy of robot pose in a multi-agent environment and additional localization methods such as scan-matching must be employed.

[9] *Pfister, Sam, Kristo Kriechbaum, Stergios Roumeliotis, and Joel Burdick.* A Weighted Range Sensor Matching Algorithm for Mobile Robot Displacement Estimation. *in Proceedings of the IEEE International Conference on Robotics and Automation, Washington D.C., May 11-15 2002*

Appendix A: Kalman Filter Source Code

```
//============================
// function Kalman
//  - Get Kalman filter estimate of robot pose
//============================
estimate_struct Kalman(double xod, double yod, double thetaod,
            double x_ns, double y_ns, double theta_ns,
            double mag1, double mag2,
            double P_state[3][3], double est_theta_old) {

  estimate_struct locdat;

  Matrix A(3,3);            //discretized matrix which in this case is identity
  Matrix C(3,3);            //measurement matrix for NS data and OD data
  Matrix Q(3,3);            //process covariance
  Matrix R(3,3);         //measurement covariance associated with NS
  Matrix Res(3,3);
  Matrix P_(3,3);           //apriori covariance matrix
  Matrix P(3,3);            //a posteriori covariance matrix
  Matrix x(3,1);           //a posteriori state estimate
  Matrix x_(3,1);       //a priori state estimate
  Matrix z(3,1);        //measurement states; all NS measurements
  Matrix K(3,3);        //Kalman Filter gains
  IdentityMatrix I(3);

  double mag;

  C = I;
  A = I;

  if(sqrt(((xod*xod)+(yod*yod))) > 1.90){
    mag = 0.001;
  }
  else if(mag1>=mag2){
    mag = mag2;
  }
  else if(mag1<mag2){
    mag = mag1;
  }

  //Enter in the initial covariance matrix
```

```
  P.Row(1) << P_state[0][0] << P_state[0][1] << P_state[0][2];
  P.Row(2) << P_state[1][0] << P_state[1][1] << P_state[1][2];
  P.Row(3) << P_state[2][0] << P_state[2][1] << P_state[2][2];


  R.Row(1) << (90/mag)*(90/mag) << 0 << 0;
  R.Row(2) << 0 << (90/mag)*(90/mag) << 0;
  R.Row(3) << 0 << 0 << (100/mag)*(100/mag);

  Q.Row(1) << (0.02)*(0.02) << 0 << 0;
  Q.Row(2) << 0 << (0.02)*(0.02) << 0;
  Q.Row(3) << 0 << 0 << (0.02)*(0.02);

//here we need to address the issue of the pi,-pi discontinuity using the
//sign of the previous estimated theta value
  if(est_theta_old > M_PI/2){  //i.e. in the 2nd quadrant
    if((theta_ns < -M_PI/2) && (thetaod > M_PI/2)){
      theta_ns = 2*M_PI+theta_ns;
      }
    if((thetaod < -M_PI/2) && (theta_ns > M_PI/2)){
      thetaod = 2*M_PI+thetaod;
      }
    }
  if(est_theta_old < -M_PI/2){ //i.e. in the third quadrant
    if((theta_ns < -M_PI/2) && (thetaod > M_PI/2)){
      thetaod = thetaod - 2*M_PI;
      }
    if((thetaod < -M_PI/2) && (theta_ns > M_PI/2)){
      theta_ns = theta_ns - 2*M_PI;
      }
    }

  x_(1,1) = xod;
  x_(2,1) = yod;
  x_(3,1) = thetaod;

  z.Row(1) = x_ns;
  z.Row(2) = y_ns;
  z.Row(3) = theta_ns;

  P_ = A*P*(A.t())+Q;
  Res = C*P_*C.t()+R;
```

```
  K = (P_*C.t())*(Res.i());
  x = x_ + K*(z-C*x_);
  P = (I-K*C)*P_;

//now we have to keep the estimated theta value within the range
//of [-pi,pi] so we do the following:
  if(x(3,1) > M_PI){
    x(3,1) = x(3,1) - 2*M_PI;
    }
  if(x(3,1) < -M_PI){
    x(3,1) = x(3,1) + 2*M_PI;
    }

  locdat.x = x(1,1);
  locdat.y = x(2,1);
  locdat.theta = x(3,1);
  locdat.P[0][0] = P(1,1);
  locdat.P[0][1] = P(1,2);
  locdat.P[0][2] = P(1,3);
  locdat.P[1][0] = P(2,1);
  locdat.P[1][1] = P(2,2);
  locdat.P[1][2] = P(2,3);
  locdat.P[2][0] = P(3,1);
  locdat.P[2][1] = P(3,2);
  locdat.P[2][2] = P(3,3);

  return(locdat);
}
```

Appendix B

Table 1: Collected Data for Wheel Odometry and Northstar

Trial 1					
Wheel Odometry			Northstar		
x	y	θ	x	y	θ
0.0000	0.0000	0.0000	0.0036	0.0034	-0.0002
0.2160	0.2160	0.7854	0.2329	0.1835	0.7048
0.4650	0.4660	0.7679	0.4890	0.4863	0.8540
0.7140	0.7160	0.7679	0.7557	0.7458	0.7707
0.9650	0.9660	0.7679	1.0172	1.0066	0.7946
1.2010	0.9960	0.0873	1.2772	1.0353	-0.0033
1.4510	1.0010	-0.0175	1.5284	1.0694	-0.0589
1.7120	1.2180	0.6981	1.7128	1.3559	1.1456
1.9640	1.4660	0.7679	2.0594	0.9713	-1.1688
1.7770	1.7080	2.1293	1.9175	1.0330	1.8758
1.5330	1.9640	2.3213	1.5064	1.7910	2.2476
1.2840	2.2140	2.3387	1.2677	1.9741	2.5884
1.0340	2.4640	2.3562	1.1314	2.2019	2.1911
0.7910	2.2760	-2.5831	1.1857	2.3696	1.2829
0.5360	2.0330	-2.3736	0.8994	2.2877	1.6748
0.2850	1.7840	-2.3911	0.5342	2.0795	2.5339
0.0350	1.5340	-2.3911	0.1574	1.8097	-2.9002
-0.2150	1.2840	-2.3736	-0.2399	1.5802	-2.7385
-0.4650	1.0340	-2.3736	-0.5435	1.3710	-2.5834
-0.7110	1.2210	2.4958	-0.7837	1.5041	2.6562
-0.9660	1.4650	2.3387	-1.0686	1.7161	2.8865
-1.2090	1.2770	-2.5482	-1.1760	1.6315	-2.6131
-1.4640	1.0340	-2.3911	-1.4365	1.4966	-2.6864
-1.4940	0.7980	-1.6930	-1.5845	1.3763	-2.5080
-1.2810	0.5380	-0.8901	-1.4134	1.1406	-1.0173
-1.0330	0.2850	-0.8029	-1.2565	0.8381	-0.9511
-0.7840	0.0360	-0.8029	-1.0701	0.5045	-0.9978
-0.5330	-0.2160	-0.8029	-0.8750	0.1200	-1.0694
-0.2840	-0.4640	-0.8029	-0.6991	-0.2022	-1.0195
-0.0340	-0.7150	-0.8203	-0.5650	-0.4877	-1.0123
0.2110	-0.5290	0.6458	-0.3296	-0.3952	0.3730
0.4640	-0.2840	0.7679	-0.0602	-0.2990	0.3913
0.7140	-0.0340	0.7679	0.2887	-0.2239	0.2327

Table 2: Kalman Filtered Data and Ground Truth

Trial 1					
Kalman Filter			Ground Truth		
x	y	θ	x	y	θ
0.0036	0.0034	-0.0002	0.0000	0.0000	0.0000
0.2328	0.1837	0.7091	0.2200	0.2200	0.7679
0.4885	0.4859	0.8404	0.4850	0.4650	0.7679
0.7527	0.7436	0.7696	0.7350	0.7200	0.7330
1.0060	0.9979	0.7769	1.0100	0.9550	0.7330
1.2529	1.0228	0.0639	1.2450	0.9700	0.0349
1.4913	1.0366	-0.0242	1.5050	0.9550	-0.0698
1.7120	1.2180	0.6981	1.7800	1.1800	0.6458
1.9640	1.4660	0.7679	2.0550	1.4100	0.6981
1.7770	1.7080	2.1293	1.8750	1.6650	2.0246
1.5330	1.9640	2.3213	1.6650	1.9500	2.2166
1.2840	2.2140	2.3387	1.4400	2.2300	2.2340
1.0340	2.4640	2.3562	1.2200	2.5150	2.2340
0.7910	2.2760	-2.5831	0.9500	2.3600	3.5605
0.5360	2.0330	-2.3736	0.6550	2.1600	3.7525
0.4232	1.9479	-2.5623	0.3600	1.9500	3.7176
0.1119	1.7073	-2.5215	0.0700	1.7450	3.7001
-0.2333	1.5018	-2.5407	-0.2300	1.5350	3.7350
-0.5242	1.2880	-2.4792	-0.5300	1.3350	3.7350
-0.7585	1.4062	2.5507	-0.7300	1.5700	2.2864
-1.0062	1.5635	2.4126	-0.9400	1.8650	2.1468
-1.1913	1.4668	-2.5612	-1.2250	1.7250	3.4907
-1.4504	1.2621	-2.4449	-1.5400	1.5600	-2.6529
-1.5197	0.9624	-1.7619	-1.6300	1.3350	-1.9548
-1.3214	0.7221	-0.9011	-1.4950	1.0300	-1.1868
-1.1533	0.5827	-0.8326	-1.3350	0.7050	-1.0996
-0.9954	0.3822	-0.8838	-1.1700	0.3800	-1.1170
-0.8294	0.0753	-0.9729	-1.0100	0.0600	-1.1170
-0.6685	-0.2216	-0.9602	-0.8500	-0.2650	-1.1170
-0.5423	-0.4973	-0.9700	-0.6900	-0.5850	-1.1345
-0.2675	-0.4105	0.5077	-0.4000	-0.4900	0.2618
-0.0383	-0.2984	0.4753	-0.0700	-0.3750	0.3316
0.2944	-0.2213	0.2831	0.2700	-0.2500	0.3142

Table 3: Collected Data for Wheel Odometry and Northstar

Trial 2					
Wheel Odometry			Northstar		
x	y	θ	x	y	θ
0.2190	0.2110	0.9076	0.2251	0.2069	0.8318
0.4640	0.4670	0.7505	0.4673	0.5305	0.8514
0.7150	0.7160	0.7679	0.7186	0.7945	0.8052
0.9640	0.9660	0.7505	0.9510	1.0710	0.8571
1.2010	0.9930	0.1222	1.2164	1.0810	0.0716
1.4520	1.0010	0.0000	1.4679	1.1270	-0.0031
1.7100	1.2220	0.6109	1.7151	1.4028	0.7246
1.9640	1.4660	0.7679	1.8791	1.6749	1.2263
1.7690	1.7050	1.9722	1.5871	1.6648	2.1639
1.5330	1.9640	2.3213	1.4198	1.9492	2.2049
1.2830	2.2140	2.3213	1.1571	2.0800	2.5998
1.0340	2.4650	2.3387	0.9606	2.2582	2.4421
0.7960	2.2680	-2.7576	0.9736	2.4232	1.6158
0.5380	2.0310	-2.4435	0.6623	2.2875	2.2700
0.2850	1.7830	-2.3911	0.3303	1.9596	-2.8989
0.0360	1.5340	-2.3736	-0.0411	1.7027	-2.8277
-0.2150	1.2830	-2.3736	-0.3340	1.4132	-2.5459
-0.4640	1.0340	-2.3736	-0.6029	1.1821	-2.5662
-0.7050	1.2300	2.7227	-0.8089	1.3189	2.9699
-0.9670	1.4640	2.3213	-1.1198	1.4830	2.9218

Table 4: Kalman Filtered Data and Ground Truth

Trial 2					
Kalman Filter			Ground Truth		
x	y	θ	x	y	θ
0.2251	0.2069	0.8318	0.2300	0.2250	0.9425
0.4672	0.5293	0.8357	0.4700	0.4900	0.8029
0.7184	0.7889	0.7903	0.7200	0.7500	0.8203
0.9537	1.0488	0.7868	0.9700	1.0100	0.8029
1.2116	1.0535	0.1089	1.2100	1.0350	0.1571
1.4605	1.0684	-0.0005	1.4650	1.0450	0.0698
1.7100	1.2220	0.6109	1.7250	1.2750	0.6632
1.9640	1.4660	0.7679	1.9800	1.5300	0.8029
1.7690	1.7050	1.9722	1.7700	1.7750	2.0071
1.5330	1.9640	2.3213	1.5400	2.0400	2.3387
1.2830	2.2140	2.3213	1.2850	2.2950	2.3213
1.0340	2.4650	2.3387	1.0350	2.5600	2.3562
0.7960	2.2680	-2.7576	0.7850	2.3550	3.5256
0.5380	2.0310	-2.4435	0.5200	2.1200	3.7874
0.3110	1.8846	-2.4599	0.2550	1.8800	3.8746
-0.0192	1.6548	-2.5304	-0.0150	1.6300	3.8921
-0.3032	1.3795	-2.4569	-0.2800	1.3900	3.8746
-0.5740	1.1513	-2.4786	-0.5550	1.1500	3.8572
-0.7842	1.2978	2.8327	-0.7850	1.3700	2.7053
-1.0411	1.4732	2.4236	-1.0400	1.6200	2.2689

Table 5: Scan-matching Data

Trial 2		
Scan-Matching		
x	y	θ
0.2251	0.2069	0.8318
0.4672	0.5293	0.8357
0.7184	0.7889	0.7903
0.9537	1.0488	0.7868
1.2116	1.0535	0.1089
1.4605	1.0684	-0.0005
1.7100	1.2220	0.6109
1.9640	1.4660	0.7679
1.7690	1.7050	1.9722
1.5330	1.9640	2.3213
1.2830	2.2140	2.3213
1.0340	2.4650	2.3387
0.7960	2.2680	-2.7576
0.5380	2.0310	-2.4435
0.3110	1.8846	-2.4599
-0.0192	1.6548	-2.5304
-0.3032	1.3795	-2.4569
-0.5740	1.1513	-2.4786
-0.7842	1.2978	2.8327
-1.0411	1.4732	2.4236