Proceedings of the SPIE Defense and Security Symposium, Unmanned Ground Vehicle Technology VI (OR54), Orlando FL, 12-16

FLEXnav: A Fuzzy Logic Expert Dead-reckoning System for the Segway RMP

Lauro Ojeda[1], Mukunda Raju[2], and Johann Borenstein [3]

[1,2,3)] The University of Michigan
Advanced Technologies Lab
1101 Beal Ave, Ann Arbor, MI 48108
[3]Ph.: 734-763-1560 (POC)
[1]lojeda@umich.edu; [2]mpraju@umich.edu; [3]johannb@umich.edu

ABSTRACT

Most mobile robots use a combination of absolute and relative sensing techniques for position estimation. Relative positioning techniques are generally known as dead-reckoning. Many systems use odometry as their only dead-reckoning means. However, in recent years fiber optic gyroscopes have become more affordable and are being used on many platforms to supplement odometry, especially in indoor applications. Still, if the terrain is not level (i.e., rugged or rolling terrain), the tilt of the vehicle introduces errors into the conversion of gyro readings to vehicle heading. In order to overcome this problem vehicle tilt must be measured and factored into the heading computation.

A unique new mobile robot is the Segway Robotics Mobility Platform (RMP). This functionally close relative of the innovative Segway Human Transporter (HT) stabilizes a statically unstable single-axle robot dynamically, based on the principle of the inverted pendulum. While this approach works very well for human transportation, it introduces as unique set of challenges to navigation equipment using an onboard gyro. This is due to the fact that in operation the Segway RMP constantly changes its forward tilt, to prevent dynamically falling over.

This paper introduces our new Fuzzy Logic Expert rule-based navigation (FLEXnav) method for fusing data from multiple gyroscopes and accelerometers in order to estimate accurately the attitude (i.e., heading and tilt) of a mobile robot. The attitude information is then further fused with wheel encoder data to estimate the three-dimensional position of the mobile robot. We have further extended this approach to include the special conditions of operation on the Segway RMP. The paper presents experimental results of a Segway RMP equipped with our system and running over moderately rugged terrain.

1. INTRODUCTION

Proprioceptive position estimation (PPE), more commonly known as dead-reckoning, is widely used for measuring the relative displacement of mobile robots.

Many conventional high-end dead-reckoning systems for ground-vehicles (typically, mobile robots) appear to be implemented according to a common approach: A 6-axes INS is fused with odometry using a Kalman Filter technique. Kalman Filters use statistical error models [Tonouchi, et al., 1994; Krantz and Gini, 1996, Baumgartner et al, 2001] to predict the behavior of sensor components. We believe that this approach is not ideal for odometry, because the statistical models can't represent well single "catastrophic" events – and the encounter of a large bump or rock is indeed catastrophic for an odometry-based system.

For many years the University of Michigan's Mobile Robotics Lab has been developing an approach that favors in-depth physical understanding of sensors and their associated error sources over the statistics-based Kalman Filter methods. Our basic philosophy is that many error mechanisms can be defined more specifically and accurately by expert reasoning. The result of our effort is a proprioceptive position estimation system called Fuzzy Logic and Expert rule-based navigation system (FLEXnav) [Ojeda and Borenstein, 2002].

The remainder of this paper is organized as follows. Section 2 describes our FLEXnav system in detail. Section 3 discusses the unique difficulties associated with the measurement of terrain pitch when using a dynamically balanced platform. Experimental results from our FLEXnav system implemented on our Segway RMP [SEGWAY] and driving over flat, horizontal terrain, as well as over moderately sloped terrain, are presented in Section 4.

2. THE FLEXNAV SYSTEM

A block diagram of the FLEXnav sensor suite is shown in Figure 1. The main inertial sensors are three single-axis fiber optic gyroscopes (FOGs) that measure rates of yaw, roll, and pitch; the two accelerometers are of lesser importance, and will be used to measure tilt on static conditions. The inertial sensors are mounted in an aluminum case on top of the Segway RMP as shown in Figure 2.

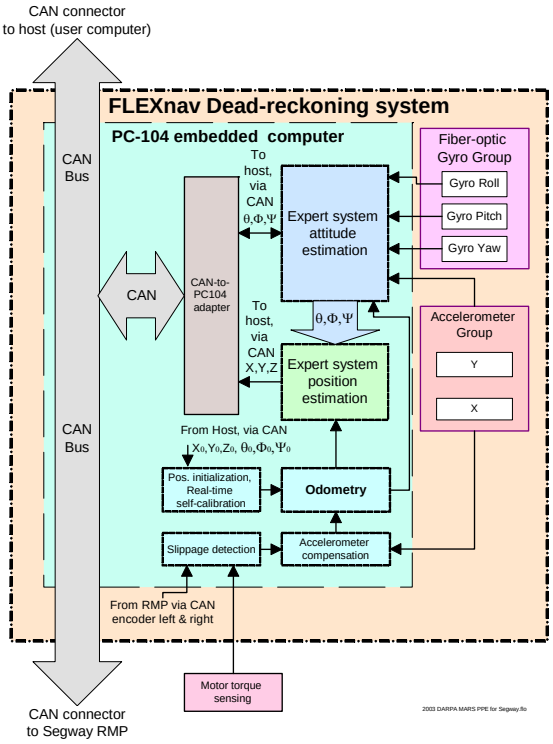


Figure 1: Block diagram of the FLEXnav dead-reckoning system for the Segway RMP.

2.1 Attitude Estimation

The attitude of a robot is a set of three angles measured between the robot's body and the absolute world coordinate system. The term "navigation frame" is used for a world coordinate system, in which the x-axis points east, the y-axis points north, and the z-axis is parallel but opposite in sign to the local gravity vector. Another coordinate system, called "body frame," can be thought of as embedded in the robot body. The body frame follows a prevailing convention used in ground vehicles for assigning the axes of coordinate systems, so that its x-axis points to the right, the y-axis points forward, and the z-axis points upward. Body axes are labeled x_b, y_b, and z_b [Kelly, 1995], and the accelerometers and gyros described in this paper were mounted in alignment with these axes. Three angles express the relative orientation between the body frame and the navigation frame, as shown in Figure 3.

The most common form of representation for these three angles is the so-called set of *Euler angles,* ϕ, θ and ψ. These three angles are called *roll* (sometimes also called "bank angle"), *pitch* (also called "elevation"), and *yaw* (also called "heading" or "azimuth"), respectively. ϕ Is the angle between x_b and the horizontal plane after rotating clockwise the pitch axis y_b through an angle θ (i.e., the plane that is normal to the z-axis of the navigation frame), θ is the angle between y_b and the horizontal plane, and ψ is the angle between x and the projection of x_b on the horizontal plane [Biezad et al., 1999]. For the mathematical treatment in the following sections we define a vector $\Lambda = [\phi, \theta, \psi]^T$ that will represent the Euler angles throughout this paper.

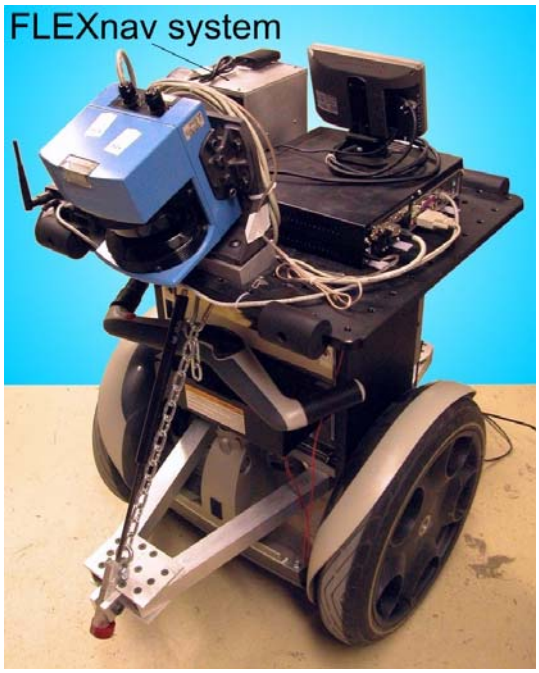


Figure 2: The University of Michigan's Segway RMP with FLEXnav PPE system installed.

Rates of rotation of the body frame relative to the navigation frame can be expressed in terms of the derivatives of the Euler angles, called "*Euler rates*."

Specifically, Euler rates Ω_ε and body rates of rotation Ω_b are related by:

$$\Omega_\varepsilon = \begin{bmatrix} \dot{\theta} & \dot{\phi} & \dot{\psi} \end{bmatrix}^T = C_b^\varepsilon \Omega_b \qquad (1)$$

where

$$C_b^\varepsilon = \begin{bmatrix} \cos\phi & 0 & \sin\phi \\ \tan\theta \cdot \sin\phi & 1 & -\tan\theta \cdot \cos\phi \\ -\dfrac{\sin\phi}{\cos\theta} & 0 & \dfrac{\cos\phi}{\cos\theta} \end{bmatrix}$$

$$\Omega_b = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T$$

ω_x, ω_y, and ω_z are the rates of rotation of the vehicle around the respective axes of the body frame.

Euler angles can be calculated from Euler rates by integrating Ω_ε over time:

$$\Lambda = \begin{bmatrix} \theta & \phi & \psi \end{bmatrix}^T = \int \Omega_\varepsilon dt \quad (2)$$

For many indoor mobile robotics applications, where floors are typically level, it is an acceptable and widely used assumption that ϕ and θ can be considered to equal zero. With this assumption we can rewrite Eq. 1:

$$\dot{\psi} = \omega_z \quad (3)$$

Table I: Technical specifications for the E-Core RA1100 and RA2100 fiber optic gyroscopes. (Courtesy [KVH])

Performance		**RA1100**	**RA2100**
Input Rate (max)	± °/sec	100	100
Resolution	°/sec	N/A	0.014
Scale factor	mV/°/sec	20	20
Nonlinearity	%, rms	<0.5	<0.5
Full Temp	%, p-p	<2	<2
Bias Stability			
Constant Temp	°/sec	0.001	0.002
Full Temp	°/sec, p-p	0.4	0.4
Angle Rand. Walk	°/hr/rt-Hz	20	5
	°/rt-hr	0.33	0.08
Bandwidth in Hz (for 3 dB with 45° phase shift)		100	100

According to Eq. 3 the heading angle ψ can be estimated by integrating only ω_z. However, on rugged terrain Eq. 3 does not hold and all three attitude parameters must be considered.

2.2 Attitude Estimation Sensors

Attitude relative to the horizontal plane *xy* (i.e., roll and pitch – ϕ and θ, respectively) is often referred to as "*tilt.*" Low cost tilt sensors are commercially available, typically in the form of electrolytic fluid sensors. However, most of these sensors are suitable only for static or quasi-static conditions because of their slow response time. In other words, they are suitable only for operation on very gently rolling terrain or slow speeds. For operation on rugged terrain faster tilt sensors are required.

For airborne applications such as missiles or aircraft, all three attitude parameters are required. The only way to obtain these parameters is through integration of rate information from sets of three mutually perpendicular highest-quality gyroscopes. In mobile robots, on the other hand, we can make use of several mitigating assumptions, that can't be made in airborne applications, such as:

1. The mobile robot is horizontal, near horizontal or at constant tilt, *most* of the time.
2. Mobile robot velocities and accelerations are small – orders of magnitude lower than those of missiles or aircraft, for which most high-quality 3-axes gyro systems are designed.
3. Wheeled and even tracked vehicles allow for the use of odometry – an advantage entirely absent in aircraft or watercraft.

Exploiting these assumptions, the FLEXnav system uses medium-priced fiber optic gyroscopes (FOGs), rather than flight-quality gyros costing tens of thousands of dollars. Specifically the FLEXnav system includes

1× KVH E-core RA2100 fiber optic gyro with analog output. We use this gyro in our system for measuring yaw rate. It costs about $3,000.

2× KVH E-core RA1100 fiber optic gyro, with analog output. We use two of these slightly lower-quality gyros to measure roll and pitch rates. They cost about $1,800 each.

From the two gyro models in our system, we use the higher quality RA2100 for the z-axis because, according to assumption (1) above, ω_z has the greatest impact on the heading angle ψ, which is the most important angle for land vehicles. We use the two somewhat noisier but less costly RA1100 FOGs to measure the less important angular rates ω_x and ω_y. Under assumption (1) (flat or moderately rolling terrain) and according to Eq. 1, ω_x and ω_y have about one order of magnitude less impact on the heading angle than ψ does. Technical specifications showing the similarities and differences between the two types of FOGs are given in Table I.

In earlier work we developed a precision calibration system for gyroscopes that reduces the errors due to the non-linearity of the scale factor and temperature by about one order of magnitude compared to an off-the-shelf unit [Ojeda et al., 2000]. We applied this same calibration technique to the FOGs in our system here.

We also incorporated two ADXL105 accelerometers made by [ANALOG DEVICES] along the X_r and Y_r axes to estimate tilt (i.e., ϕ and θ). This can be done when the robot is static or moving linearly at constant speed (a similar sensor configuration was used in [Rehbinder and Hu, 2001]). Under these conditions tilt can be calculated as:

$$\phi = sin^{-1}\left(\frac{g_x}{g\cos\theta}\right) \quad (4)$$

$$\theta = sin^{-1}\left(\frac{g_y}{g}\right) \quad (5)$$

where

g – gravitational acceleration

$g_{x/y}$ – x/y-component of the gravitational acceleration

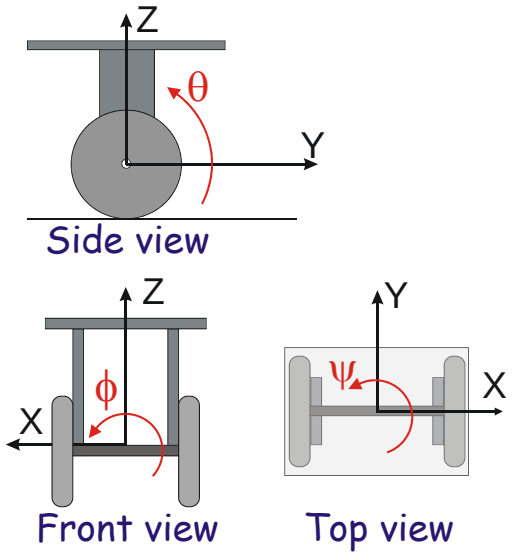


Figure 3: Robot axes and Euler angles

Similar to gyroscopes, accelerometers suffer from bias drift problems. It is well established that accelerometers are generally not suitable for measuring linear displacement in mobile robots [Barshan and Durrant-Whyte, 1995]. This is because accelerometer measurements must be integrated twice to yield position, and thus even small amounts of drift will grow substantially and without bound. However, when the vehicle is static or moving at constant speed, accelerometers can be used as tilt sensors since tilt information can be derived directly from the accelerometer readings, according to Eqs. 4 and 5. No integration is needed and therefore drift is not a dominant source of errors. Rather, other error sources become relatively more significant, such as inaccuracy, noise, non-linearity, and sensitivity to vibration. Nonetheless, accelerometers can be useful to bound and reset the tilt information calculated by the gyros.

Under dynamic conditions, i.e., when the robot accelerates, accelerometers will also measure the acceleration of the robot in addition to the robot's tilt. This ambiguity can be resolved using encoder readings.

2.3 Static bias drift compensation

One well known source of errors in gyroscopes is the static bias drift. In this paper we are not focusing on this (significant) error source because a trivial short-term calibration method is effective in dealing with this problem: Prior to each mobile robot mission (i.e., while the robot is standing still) the gyro output is sampled for, say, 5-10 seconds, and the readings are averaged. Then, once the robot is moving, this averaged static bias value is subtracted from all subsequent gyro readings. Because of the relatively low static bias drift in FOGs, mission durations on the order of 10 minutes are feasible before a new static bias drift value should be determined by repeating the above short-term calibration. This or similar procedures are widely used; we include this brief description for completeness.

One should also note that this short-term calibration procedure also compensates for the constant planetary rotation – provided the robot does not spend significant amounts of time tilted. While the robot remains tilted, planetary rotation affects all gyros differently from the way they were affected during the static bias drift calibration phase, thereby introducing new errors.

2.4 Fuzzy Data Fusion Using Expert Rules

Based on the specific physical shortcomings and strengths of each sensor modality, we defined the following basic expert rules:

Rule 1: If the vehicle is in the process of turning about any of its axes, our best attitude estimate is the one derived from the gyroscope outputs, that is $\Lambda \approx \Lambda_g$, where

$$\Lambda_g = \begin{bmatrix} \theta_g & \phi_g & \psi_g \end{bmatrix}^T \quad (6)$$

and the index '*g*' indicates that the value was derived from gyro data. ϕ_g, θ_g and ψ_g are computed from gyro data according to Eqs. 1 and 2.

Rule 2: If the robot is not turning around any axis and is not accelerating linearly, the accelerometers can directly measure the roll and pitch attitude parameters (ϕ_a and θ_a). If the conditions of Rule 2 are met for several seconds uninterruptedly, then we can also measure and correct for the bias drift errors of the gyroscope (see [Ojeda et al., 2000]) and reset the tilt parameters of the robot to the tilt estimated by the accelerometers (see Eqs. 4 an 5), therefore

$$\phi \approx \phi_a \text{ and } \theta \approx \theta_a \quad (7)$$

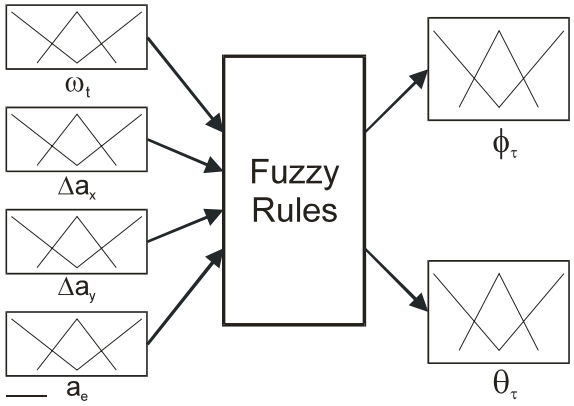


Figure 4: Inputs and outputs of the FLEXnav system.

where we define the symbol "$\approx$" as meaning "*weighted toward*." The meaning of this term will become evident in the discussion on Fuzzy Logic, below.

Even though the sensor integration conditions are well defined and sensible, it is not feasible to implement them as strictly binary rules. This is due to the natural imprecision of the sensors and because conditions like "robot not turning" or "constant speed" are not clearly defined when the vehicle is in motion on rugged terrain.

A fusion algorithm that takes into account the physical capabilities and limitations of each sensor is therefore necessary. We found that Fuzzy Logic is well suited for this task.

- Fuzzy Logic uses rules to map inputs and outputs. Using natural language, expert rules such as the ones described above can be translated easily into the IF-THEN statements used by Fuzzy Logic rules.
- Fuzzy Logic is specifically designed to deal with the imprecision associated with noisy sensors.
- Trying to use a deterministic approach to solve this kind of problem would require the development of a highly non-linear system model, which, in turn, would increase the complexity and development time. Fuzzy Logic, on the other hand, can handle nonlinear models of arbitrary complexity [Jang et al., 1997].

Our fuzzy data fusion uses four fuzzy membership function[1] inputs and two outputs, as shown in Figure 4. The first input represents the state of rotation (i.e., whether the platform is rotating about any axis). The parameter that determines this condition is calculated by:

$$\omega_t[k] = |\omega_x[k]| + |\omega_y[k]| + |\omega_z[k]| \quad (8)$$

where k is an integer that counts the sampling intervals T_s since the beginning of the experiment.

The second and third input use accelerometer data to determine if the acceleration of the robot is changing:

$$\Delta a_x[k] = a_x[k] - a_x[k-1] \quad (9)$$

$$\Delta a_y[k] = a_y[k] - a_y[k-1] \quad (10)$$

$\Delta a_x \approx 0$ and $\Delta a_y \approx 0$ means that the robot is either standing or moving with constant acceleration, and that it

Table II: Fuzzy logic rules used in the FLEXnav system.

$\Re_1$: ***if*** ω_t is not SLOW ***then*** ϕ_τ is GYRO ***and*** θ_τ is GYRO
$\Re_2$: ***if*** Δa_x is HIGH ***then*** θ_τ is GYRO
$\Re_3$: ***if*** Δa_y is HIGH ***then*** θ_τ is GYRO
$\Re_4$: ***if*** Δa_e is HIGH ***then*** ϕ_τ θis GYRO ***and*** θ_τ is GYRO
$\Re_5$: ***if*** ω_t is SLOW ***and*** Δa_x is LOW ***and*** a_e is LOW ***then*** ϕ_τ is ACCEL
$\Re_6$: ***if*** ω_t is SLOW ***and*** Δa_y is LOW ***and*** a_e is LOW ***then*** θ_τ is ACCEL
$\Re_7$: ***if*** ω_t is SLOW ***and*** Δa_x is LOW ***and*** a_e is MED ***then*** ϕ_τ is BOTH
$\Re_8$: ***if*** ω_t is SLOW ***and*** Δa_y is LOW ***and*** a_e is MED ***then*** θ_τ is BOTH
$\Re_9$: ***if*** ω_t is SLOW ***and*** Δa_x is MED ***and*** a_e is LOW ***then*** ϕ_τ is BOTH
$\Re_{10}$: ***if*** ω_t is SLOW ***and*** Δa_y is MED ***and*** a_e is LOW ***then*** θ_τ is BOTH
$\Re_{11}$: ***if*** ω_t is MED ***and*** Δa_x is LOW ***and*** a_e is LOW ***then*** ϕ_τ is BOTH
$\Re_{12}$: ***if*** ω_t is MED ***and*** Δa_y is LOW ***and*** a_e is LOW ***then*** θ_τ is BOTH
$\Re_{13}$: ***if*** ω_t is MED ***and*** Δa_x is not LOW ***then*** ϕ_τ is GYRO
$\Re_{14}$: ***if*** ω_t is MED ***and*** Δa_y is not LOW ***then*** θ_τ is GYRO
$\Re_{15}$: ***if*** ω_t is MED ***and*** Δa_e is not LOW ***then*** ϕ_τ is GYRO ***and*** θ_τ is GYRO

[1] In Fuzzy Logic, a "membership function" is defined as a curve that maps each point in the input space to a membership value or grade between 0 and 1.

is standing or moving on terrain that has a constant slope. The "standing-or-moving" ambiguity is resolved using encoder information, which is the fourth input to the system. The first derivative of the velocity as measured by the encoders represents the rover's acceleration

$$a_e[k] = \frac{\Delta v_e[k]}{T_s} = \frac{v_e[k] - v_e[k-1]}{T_s} \tag{11}$$

The outputs of the fuzzy fusion system, ϕ_τ and θ_τ, are dimensionless weighting factors that emphasize either the gyroscope readings, the accelerometer readings, or weigh one relative to the other. In practice the maximum range for these weighting factors are chosen by the system designer depending on the accuracy and noise specification of the sensors.

Based on the Fuzzy Logic rules above, we can now compute the attitude

$$\phi[k] = \phi_g[k] + (\phi_a[k] - \phi_g[k])\,\phi_\tau[k] \tag{12}$$

$$\theta[k] = \theta_g[k] + (\theta_a[k] - \theta_g[k])\,\theta_\tau[k] \tag{13}$$

from Eq. 2:

$$\phi_g[k] = \phi[k-1] + \dot{\phi}_g[k]T_s \tag{14}$$

$$\theta_g[k] = \theta[k-1] + \dot{\theta}_g[k]T_s \tag{15}$$

where T_s is the sampling period, and from Eq. 4 and Eq 5:

$$\theta_a[k] = sin^{-1}\left(\frac{g_y[k]}{g}\right) \tag{16}$$

$$\phi_a[k] = sin^{-1}\left(\frac{g_x[k]}{g\,cos(\theta[k-1])}\right) \tag{17}$$

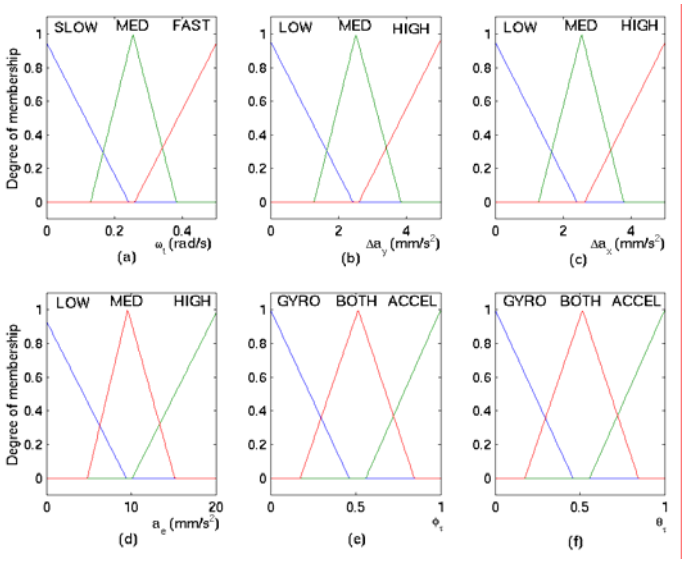


Figure 5: Membership functions of the fuzzy fusion system.

Once the inputs and outputs are identified and defined, the relationship between them must be established. As mentioned above, Fuzzy Logic uses *if-then* rules to map inputs and outputs. For our Fuzzy Logic fusion system we translated our knowledge base (fusion rules) into the fuzzy rules shown in Table II**.** The membership functions used as input and output of our system are also shown in Figure 5.

It should be noted that the FLEXnav system described up to this point only reduces errors in roll and pitch (ϕ and θ, respectively). It is important to reduce these errors, since on rugged terrain roll and pitch affect the computation of the heading angle ψ, according to Eq. 1.

3. THE SEGWAY RMP ON 3-D TERRAIN

When using the Segway RMP on sloped terrain, there exists one problem that is unique to this platform: The measurement of terrain inclination (also referred to as terrain pitch in the remainder of this paper). On conventional four-wheeled robots without a suspension system, the pitch of the robot is identical to the pitch of the terrain – at least on moderately rugged, rolling terrain. The Segway RMP, however, changes pitch as a function of its dynamic stabilization system. Indeed, the angle of forward or backward pitching of the Segway depends on a variety of factors, such as terrain characteristics, weight distribution of the payload, acceleration of the robot, etc. As a result the pitch of the robot does not correspond to the pitch of the terrain. Any inertial navigation system mounted rigidly on the Segway can thus only measure the pitch of the platform relative to the horizon (or relative to the gravity vector), but not the

pitch of the terrain. Yet, it is the pitch of the terrain that must be known in order to compute correctly the displacement of the robot on the X-Y plane.

3.1 Range Sensor-based Terrain Inclination

In order to measure the terrain pitch Θ we decided to measure platform pitch (relative to the horizon and denoted θ in Eqs. 1-16) with the FLEXnav system, and subtract from that measurement the pitch of the platform relative to the terrain (we refer to this relative pitch as the "platform/terrain pitch," denoted β). Measuring platform/terrain pitch, however, is no trivial matter. Indeed, we found no elegant solution and ended up employing a not-so-elegant one: We installed downward looking IR range sensors (model GP2D12 made by [SHARP]) underneath the custom bumpers of our Segway RMP, as shown in Figure 6.

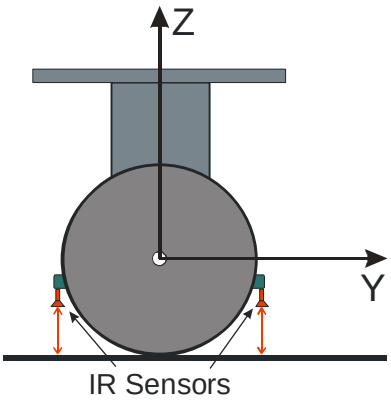


Figure 6: Two infrared range sensors are mounted underneath the Segway to measure the relative angle between the robot and the terrain.

Even though theoretically only one IR sensor is necessary to estimate β, it is preferable to use two sensors, one mounted in the back and one in front of the body. This arrangement is preferable because:

- This approach eliminates ambiguities in cases when one of the sensors sees a change of inclination before the wheels reach it.
- The IR sensors' output response is non linear and their resolution worsens with distance. By using two sensors, we can guarantee that at least one of them is working in the higher-accuracy close range.
- On some terrains such as sand, grass or thick carpet, the measured range does not correspond to the distance between the sensor and the point of contact of the wheel with the floor. However, we can still get a good estimation of the terrain inclination by combining the front and rear range measurement (as will be shown below by Eq. 22).

In the remainder of this section we use the following symbol convention: any parameter estimated based on a particular sensor has the name of that sensor as an index.

β_R, β_F – platform/terrain pitch as estimated by the Rear and Front IR sensor, respectively;

β_{RF} – estimated pitch between the plane of the Rear/Front sensor and the terrain

The relative pitch between the platform and the terrain for the front (β_F) and rear (β_R) IR sensors, can be derived from the geometric relations shown in see Figure 7

Using

$$R_{IR}\cos\beta_R - H = R_D \sin\beta_R \tag{18}$$

we obtain

$$\cos\beta_R = \frac{R_{IR}H + \sqrt{H^2 R_{IR}^{\ 2} - (R_{IR}^{\ 2} + R_D^{\ 2})(H^2 - R_D^{\ 2})}}{R_{IR}^{\ 2} + R_D^{\ 2}} \tag{19}$$

and using

$$H - F_{IR}\cos\beta_F = F_D \sin\beta_F \tag{20}$$

we obtain

$$\cos\beta_F = \frac{F_{IR}H + \sqrt{H^2 F_{IR}^{\ 2} - (F_{IR}^{\ 2} + F_D^{\ 2})(H^2 - F_D^{\ 2})}}{F_{IR}^{\ 2} + F_D^{\ 2}} \tag{21}$$

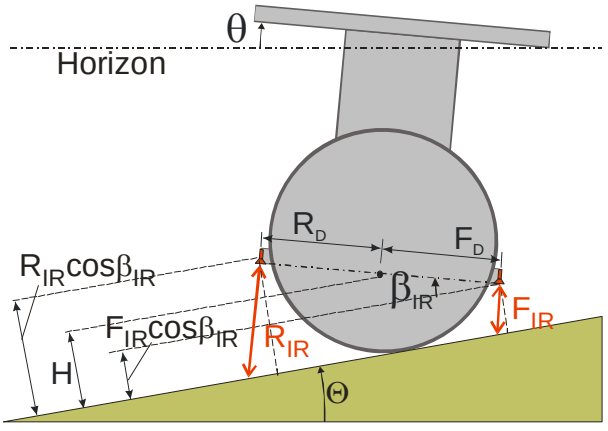


Figure 7: Range sensor location and corresponding angles.

where

R_{IR}, F_{IR} – distance to ground measured by the Rear/Front IR sensor;

H – height of the axis of rotation of the plane of the sensors;

R_D, F_D – distance between the Rear/Front IR sensor and the sensor axis of rotation

By simple geometry, we obtain a third solution for the pitch angle β_{RF}:

$$\tan \beta_{RF} = \frac{R_{IR} - F_{IR}}{R_D - F_D} \tag{22}$$

The plane of the sensors however is not necessarily located in the same plane as the center of rotation of the wheels; hence it also swings when the Robot pitches forward or backward. This results in a varying height of the plane of sensors from the ground (see Figure 8), which can be accounted for as follows:

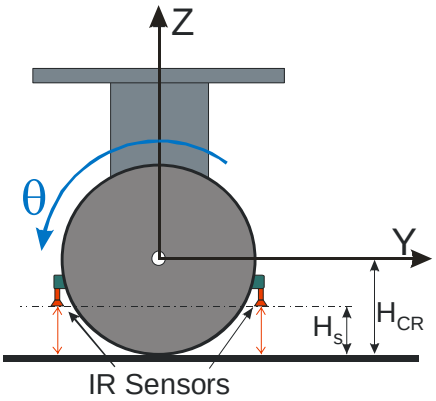


Figure 8: Relationship between the wheel center of rotation and the sensor plane.

$$H = H_{CR}(1 - \cos\theta) + H_S \cos\theta \tag{23}$$

Depending on the terrain conditions (as described above), the final platform/terrain pitch angle based on the infrared range sensors β_{IR}, can be estimated using either Eq. 19, 21, or 22. With this value the terrain inclination can be computed using (see Figure 9):

$$\Theta = \beta_{IR} - \theta \tag{24}$$

3.2 Self Calibration Parameter Estimation

The accuracy of this technique depends on the accuracy of measuring the exact location of the range sensors with respect to the center of rotation H_{CR}, which could be difficult since the wheels' center of rotation may not be easy to measure.

Our algorithm is based on the fact that on flat terrain the pitch of the robot as seen by FLEXnav, θ, and the pitch estimated using the range sensors, β_{IR}, should be the same. Considering this, we can run a short procedure having the robot pitch back and forth in place on flat terrain, while collecting sensor data.

For the rear sensor, we can rearrange Eq. 18 and Eq. 23 (a similar treatment can be applied to the front IR sensor, but we skip the description of that process here).

$$H_{CR}(1 - \cos\theta) + H_S \cos\theta + R_D \sin\beta_R = R_{IR} \cos\beta_R \tag{25}$$

Assuming $\theta = \beta_R$, we can obtain a linear system of equations as follows:

$$A = \begin{bmatrix} 1-\cos\theta_1 & \cos\theta_1 & \sin\theta_1 \\ 1-\cos\theta_2 & \cos\theta_2 & \sin\theta_2 \\ . & . & . \\ . & . & . \\ 1-\cos\theta_n & \cos\theta_n & \sin\theta_n \end{bmatrix} \tag{26}$$

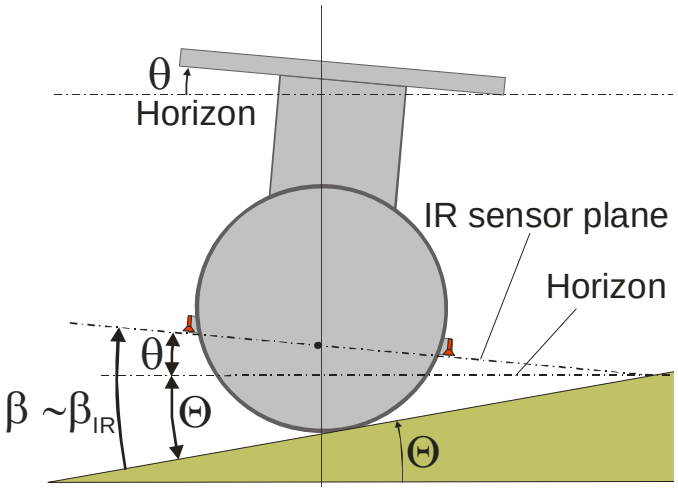


Figure 9: Definition of angles for computing terrain inclination.

$$B = \begin{bmatrix} R_{IR1}\cos\theta_1 \\ R_{IR2}\cos\theta_2 \\ . \\ . \\ R_{IRn}\cos\theta_n \end{bmatrix} \quad (27)$$

$$x = \begin{bmatrix} H_{CR} \\ H_S \\ R_D \end{bmatrix} \quad (28)$$

$$Ax = B \quad (29)$$

In $n = 3$, we can solve Eq. 29 finding the inverse of matrix A and applying Eq. 30:

$$x = A^{-1}B \quad (30)$$

However, considering that the sensor data is noisy we use a large number of samples (n>>3). Solving Eq. 29 in the least squares sense we can get a good approximation of H_{CR}, H_S and R_D.

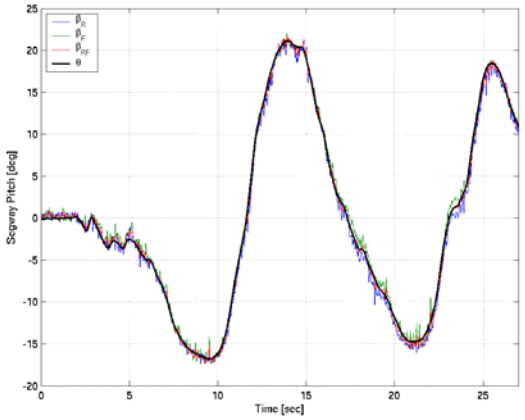


Figure 10: Pitch of the robot on flat terrain estimated using FLEXnav (θ), and by the IR sensors (β) after parameter calibration

The same procedure can be repeated using Eqs. 20 and 23 in order to find the value F_D. After computing these constants we can verify them by estimating the platform/terrain pitch of the robot on flat terrain, using Eq. 19, 21 or 22; and comparing it with the pitch of the robot estimated by FLEXnav as shown in Figure 10.

4. EXPERIMENTAL RESULTS

In the first part of this section, we report on results obtained with the Segway running only on flat, horizontal terrain. We refer to this as 2-D terrain. In the second part of this section we present our results of runs on sloped 3-D terrain.

4.1 FLEXnav on Segway: 2-D Terrain

We conducted experiments on three different 2-D terrain surfaces: (1) indoors on carpet, (2) outdoors on an asphalted parking lot, and (3) outdoors on grass.

In each of the three experimental environments we ran the Segway under joystick control on a near-rectangular, closed-loop path, so that it stopped exactly where it started. In each experiment we performed multiple runs in clockwise (cw) and counter-clockwise (ccw) direction. Upon return to the starting position we measured the discrepancy between the actual robot position and the position reported by the FLEXnav PPE. The resulting discrepancies are the position errors produced by the FLEXnav PPE. For each experiment, we provide a plot that shows the x- and y-values of the final position error for each individual run. We also provide for each experiment a set of values, X_e and Y_e. These values represent the *average of absolute errors (AAE)* in X-and Y-direction, respectively. For example, if for a certain experiment we performed n runs and measured position errors (x_1, y_1), $(x_2, y_2)\ldots$ (x_n, y_n) at the end of each run, then,

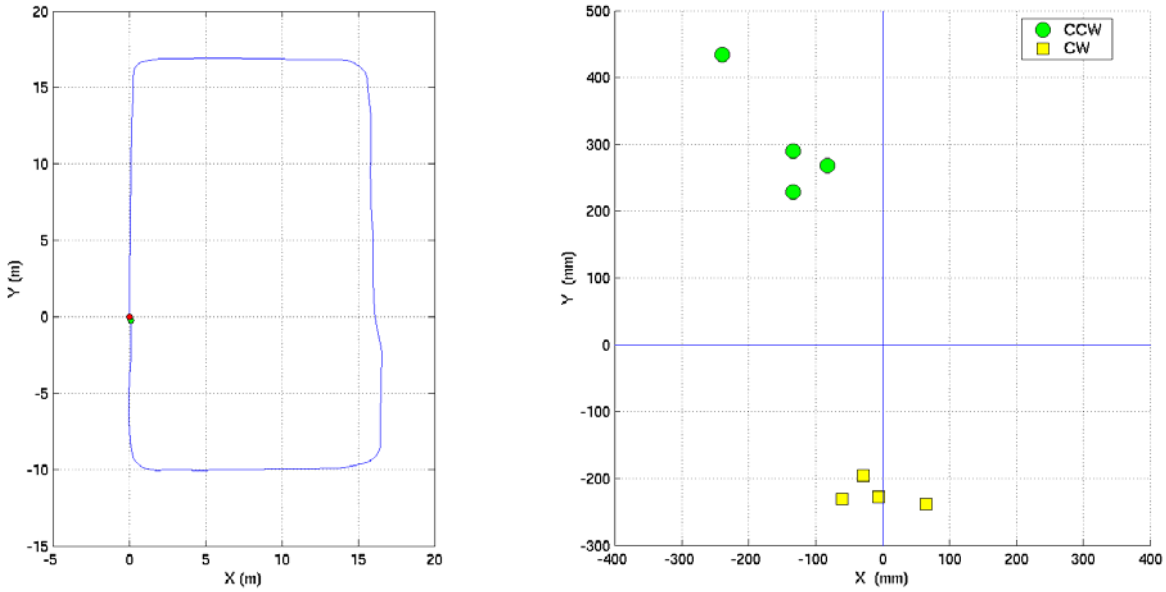
$$X_e = \frac{1}{n}\sum_{i=1}^{n}|x_i| \text{ and } Y_e = \frac{1}{n}\sum_{i=1}^{n}|y_i| \quad (31)$$

Experiment 1: On carpet (see Figure 12)

Average speed: 40 cm/sec, total traveled distance: ~85 m

AEE ccw: X_e = 147 mm; Y_e = 305 mm => 0.40% of distance traveled

AEE cw: X_e = 40 mm; Y_e = 223 mm => 0.27% of distance traveled



a **b**

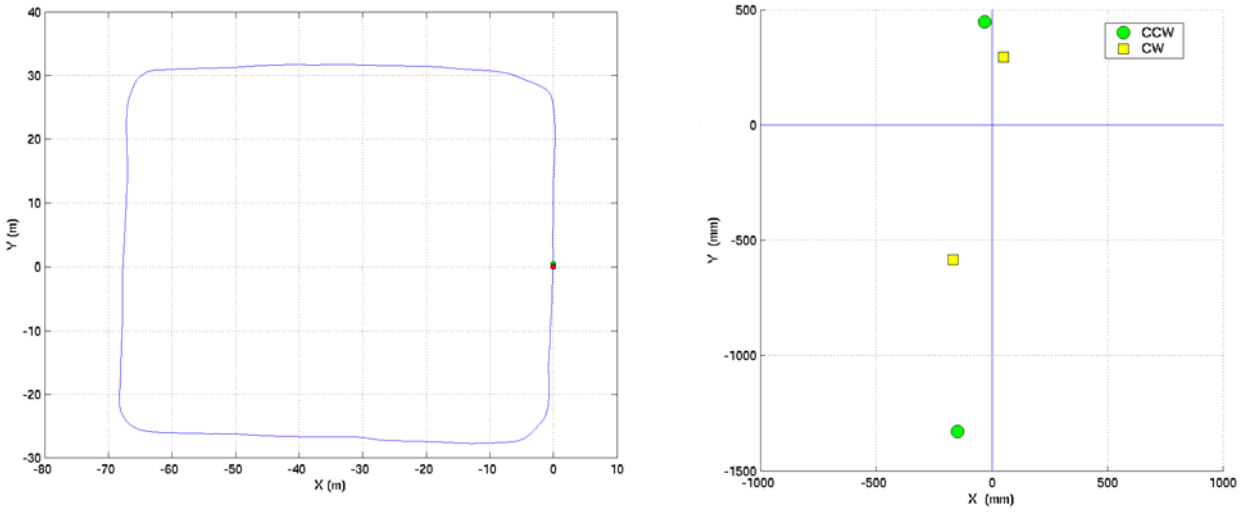
Figure 11: Experiment on carpet. a. Trajectory. b. Return position errors.

Experiment 2: On asphalt (see Figure 12)

Average speed 60 cm/sec; total travel distance: ~250 m

AEE ccw: X_e = 90 mm; Y_e = 888 mm => 0.35% of distance traveled

AEE cw: X_e = 09 mm; Y_e = 440 mm => 0.18% of distance traveled



a **b**

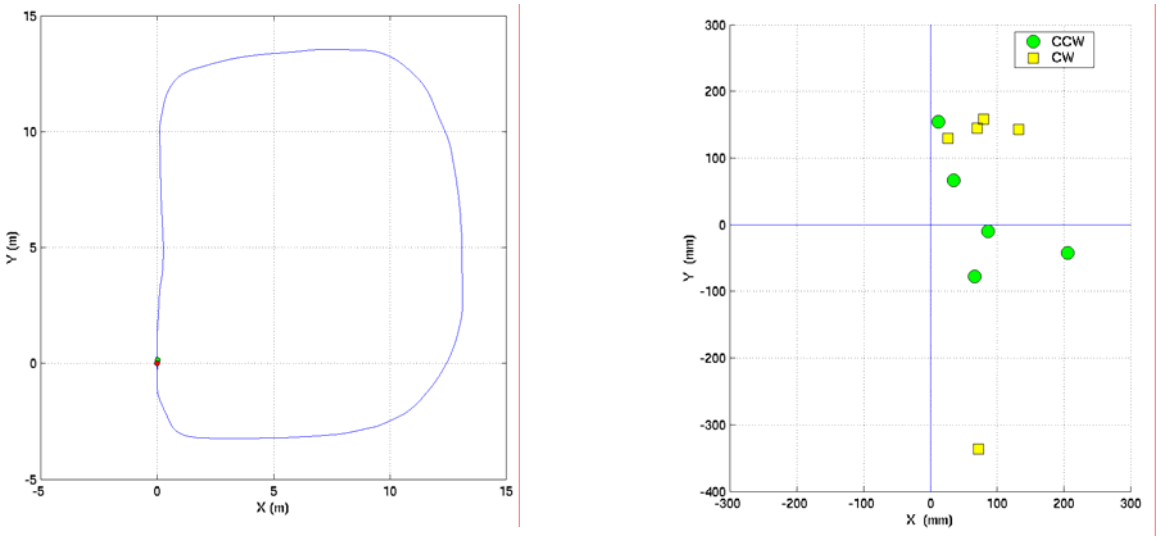
Figure 12: Experiment on asphalt. a. Trajectory. b. Return position errors.

Experiment 3: On grass (somewhat rough ride) (see Figure 13)

Average speed 50 cm/sec; total travel distance: ~55 m

AEE ccw: X_e = 81 mm; Y_e = 70 mm => 0.19% of distance traveled

AEE cw: X_e = 76 mm; Y_e = 182 mm => 0.36% of distance traveled



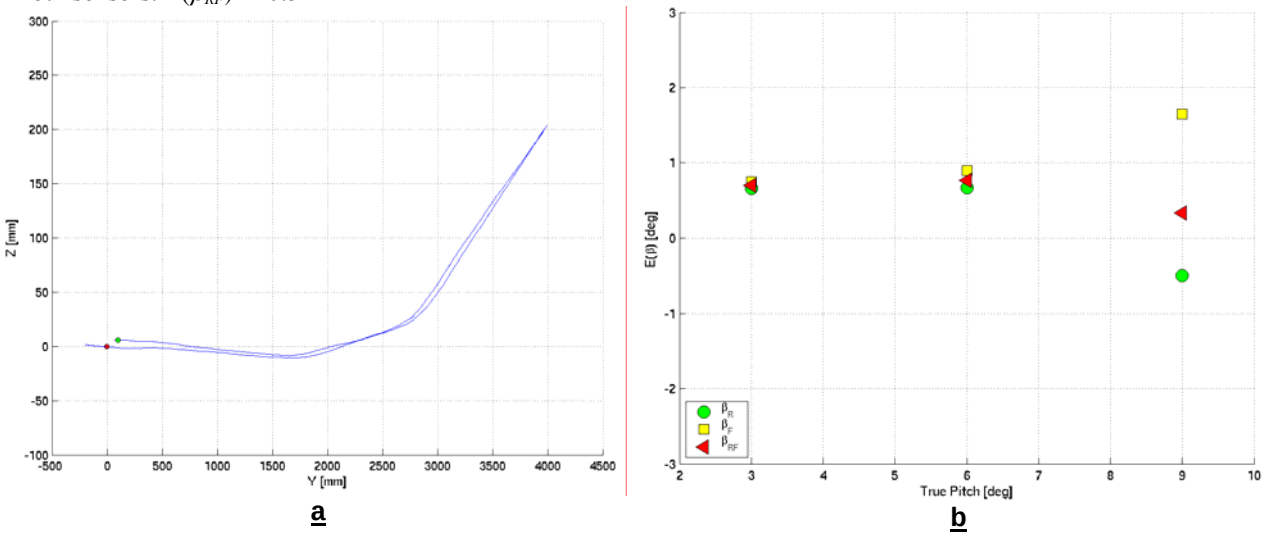
a **b**

Figure 13: Experiment on grass. a. Trajectory. b. Return position errors.

4.2 FLEXnav on Segway: 3-D Terrain

In order to test the IR range sensor-based terrain estimation algorithm, we performed several experiments on smooth 3-D terrain. We started each run on flat terrain. We then steered the robot up an incline and then back down to the original position (see Figure 14a). This procedure was performed three times for each inclination (3, 6 and 9 degrees), after each experiment, the average pitch error was computed and compared with the true platform inclination (see Figure 14b). For each experiment we computed the terrain inclination based on the rear (β_R), front (β_F), and both (β_{RF}) infrared sensors. The average error in estimating the terrain inclination, E, for each method was:

Rear sensor: $E(\beta_R) = 0.6°$
Front sensor: $E(\beta_F) = 1.1°$
Both sensors: $E(\beta_{RF}) = 0.5°$



a **b**

Figure 14: Experiment on variable-angle tilt platform.
a. Trajectory of the robot going up and down the ramp (9°), b. average pitch angle error.

The following experiments were performed in similar fashion as the 2-D experiments. We remote-controlled the Segway to follow a near-rectangular, closed-loop path. Along the Y-axis of the trajectory, on one side of the rectangular path, we placed a double-sided ramp, which created a 3-D environment (see Figure 15a). The robot had to complete four laps going up and down the ramp before returning to the starting point. The final position errors without and with the range sensor terrain inclination algorithm are shown in Figure 15b. The slope of the ramp on both sides was about 9°, the average speed for these experiments was 200 mm/sec, and the total traveled distance 75 meters.

Since the ramp was placed along the Y-axis the most significant error can be expected to be in the Y-direction.

Before applying IR range sensor correction:

AEE ccw: $X_e = 65$ mm; $Y_e = 217$ mm
=> 0.30% of distance traveled

AEE cw: $X_e = 132$ mm; $Y_e = 285$ mm
=> 0.42% of distance traveled

After applying IR range sensor correction:

AEE ccw: $X_e = 82$ mm; $Y_e = 37$ mm
=> 0.12% of distance traveled

AEE cw: $X_e = 104$ mm; $Y_e = 56$ mm
=> 0.16% of distance traveled

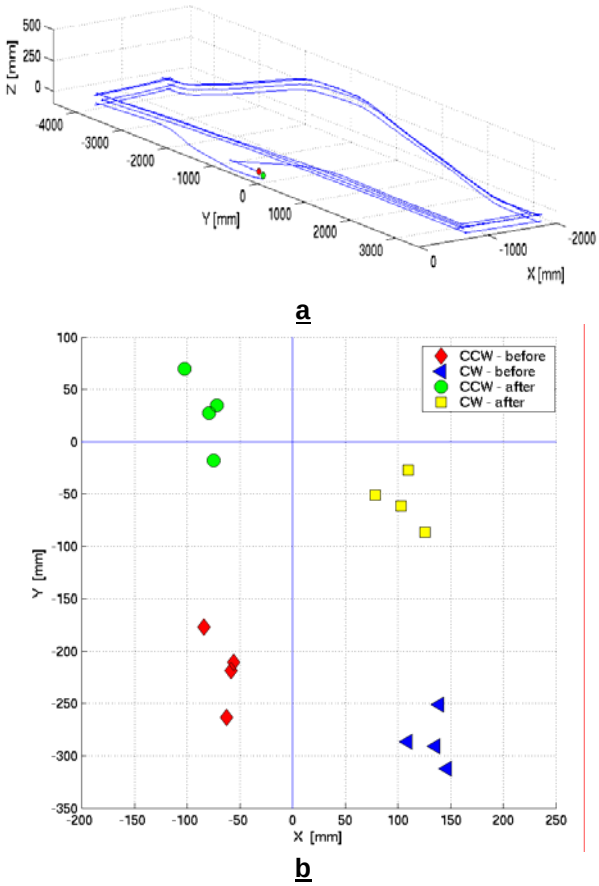


Figure 15: Experiment on 3D terrain. a. Trajectory. b. Return position errors, before and after using infrared range sensor-based terrain inclination estimation.

5. CONCLUSIONS

On rugged terrain momentary tilt information must be taken into account for correcting heading measurements, regardless of the quality of the main heading sensor (a fiber optic gyro, in our case here). To do so we developed and implemented a Fuzzy Logic expert rules-based navigation system, called FLEXnav.

The Segway RMP raises a unique problem for proprioceptive position estimation: It is not possible to distinguish between the (constantly fluctuating) pitch of the robot and the actual terrain inclination. As a result, the FLEXnav proprioceptive pose estimation system (or any other Inertial Measurement Unit (IMU), for that matter), is unable to measure the terrain inclination. Consequently, odometry-based measurements of linear displacement are correct only on flat, horizontal terrain, but not if the robot traveled on a slope.

To overcome this problem we developed an IR range sensor-based system that allows estimation of the relative angle between the Segway platform and the terrain. Continuous measurement of this so-called platform/terrain pitch allows our system to estimate the terrain inclination.

We feel that the current IR range sensor-based system is not a very elegant solution, although it is practical and produced good experimental results. We are currently working on another approach, one that measures energy consumption based on motor current measurements, in order to estimate vertical displacement and thus, terrain pitch. This method, however, is not yet sufficiently developed to allow inclusion in this paper.

Acknowledgements

This work was funded by the U.S. Department of Energy under Award No. DE-FG04-86NE3796 and by DARPA under Grant #F007571.

6. REFERENCES

Barshan, B. and Durrant-Whyte, H.F., 1995, "Inertial Navigation Systems Mobile Robots." *IEEE Transactions on Robotics and Automation*, Vol. 11, No. 3, June 1995, pp. 328-342.

Biezad, D., 1999, "Integrated Navigation and Guidance Systems." *American Institute of Aeronautics and Astronautics, AIAA Education Series*

Borenstein, J. and Koren, Y., 1987, "Motion Control Analysis of a Mobile Robot." *Transactions of ASME, Journal of Dynamics, Measurement and Control*, Vol. 109, No. 2, pp. 73-79.

Borenstein, J. and Feng. L., 1996, "Measurement and Correction of Systematic Odometry Errors in Mobile Robots." *IEEE Transactions on Robotics and Automation*, Vol. 12, No. 6, Dec., pp. 869-880

Jang, J.-S., Sung, C.-T. and Mizutani, E., 1997, "Neuro-Fuzzy and Soft Computing: A computational Approach to Learning and Machine Intelligence" *Matlab Curriculum Series*

Janosi, B. Hanamoto, 1961, "Analytical Determination f Drawbar Pull as a Function of Slip for Tracked Vehicles in Deformable Soils." *Proc. of the First Int. Conf. On Terrain-Vehicle Systems*, Edizioni Minerva Tecnica, Torino.

Kelly, A., 1995, "An Intelligent, Predictive Control Approach to High-Speed Cross-Country Autonomous Navigation Problem." *Ph.D. Thesis. The Robotics Institute, Carnegie Mellon University*, Pittsburgh, PA.

Krantz, D. and Gini, M., 1996, "Non-Uniform Dead-Reckoning Position Estimate Updates." *Proc. of the 1996 IEEE Int. Conf. on Robotics and Automation Minneapolis*, MN, April 22-25, pp. 2061-2066.

Measurement Specialties, Inc. Fairfield, NJ 07004, http://www.msiusa.com.

Ojeda, L. Chung, H., and Borenstein, J., 2000, "Precision-calibration of Fiber-optics Gyroscopes for Mobile Robot Navigation." *Proc. of the 2000 IEEE Int. Conf. on Robotics and Automation, San Francisco*, CA, pp. 2064-2069.

Ojeda, L. and Borenstein, J., 2002, "FLEXnav: Fuzzy Logic Expert Rule-based Position Estimation for Mobile Robots on Rugged Terrain." *Proc. of the 2002 IEEE Int. Conf. on Robotics and Automation*. Washington DC, USA, 11 - 15 May, pp. 317-322.

Ojeda, L. and Borenstein, J., 2004, "Methods for the Reduction of Odometry Errors in Over-constrained Mobile Robots." Accepted for publication in *Autonomous Robots*.

Rehbinder, H., Hu, X., 2001, "Drift-free attitude estimation for accelerometer rigid bodies." *Proc. of the 2001 IEEE Int. Conf. on Robotics and Automation,* Seoul, Korea, May 21-26, 2001, pp. 4244-4249.

Tonouchi, Y., Tsubouchi, T., and Arimoto, S., 1994, "Fusion of Dead-reckoning Positions with a Workspace Model for a Mobile Robot by Bayesian Inference." *Int. Conf. on Intelligent Robots and Systems (IROS '94),* Munich, Germany, September 12-16, pp. 1347-1354.

Companies

Analog Devices, Inc., Norwood, MA 02062, USA, http://www.analog.com.

KVH Industries, Inc., 8412 W. 185th St., Tinley Park, IL 60477, USA, http://www.kvh.com.

Segway LLC, 286 Commercial St., Manchester, NH 03101, USA, http://www.segway.com.

Sharp Electronics Corp., 1300 Naperville Drive, Romeoville, IL 60446, USA http://sharp-usa.com.