Proceedings of the 2004 IEEE
International Conference on Robotics & Automation
New Orleans, LA • April 2004

Robust Mobile Robot Localization using Optical Flow Sensors and Encoders

Sooyong Lee
Department of Mechanical and System Design Engineering
Hongik University
Seoul, Korea
sooyong@hongik.ac.kr

Jae-Bok Song
Department of Mechanical Engineering
Korea University
Seoul, Korea
jbsong@korea.ac.kr

Abstract— **Open-loop estimation methods are commonly used in mobile robot applications. Their strength lies in the speed and simplicity of an estimate. However, these methods can sometimes lead to inaccurate or unreliable positional estimates. Using one or more optical flow sensors, a method has been developed which can accurately track position in both ideal kinematic conditions and otherwise. Using optical flow techniques and available sensors, reliable positional estimates are made. The sensor provides accurate measurement of the movement at the sensor location. Even though the sensor does not provide angular displacement, the robot movement is estimated using only one sensor even with wheel slip. However, when the robot moves sideways due to external disturbance, redundant sensors are used in order to estimate the configuration of the robot. Pseudo inverse based estimation and the extended Kalman filter based estimation are presented to show the effectiveness of the proposed approach. Location of the sensors has also been investigated in order to minimize errors caused by inaccurate sensor readings. Finally, the method is implemented and tested using a potential field based navigation scheme.**

I. Introduction

Accurate position estimation is a key component to the successful operation of most autonomous mobile robots. In general, there are three phases that comprise the motion of a mobile robot: localization, path planning, and path execution. During localization, the position and orientation in the reference coordinate system is determined using external sensors. A path is then planned that passes through a goal point or a series of intermediate via points. The final phase is the execution of the planned path. This process is repeated so that the robot will remain on course towards the goal.

Commonly, open-loop estimation (dead-reckoning) is used for intermediate estimation of position during path execution. Open-loop estimation is used because the encoders are available for motor control, which provides actual angular displacement of the wheels. However, due to errors in kinematic model parameters or wheel slip, poor position estimates may occur. Poor estimates in position during path execution require more frequent localizations to be made, incurring extra overhead and possibly slowing the movement of the robot. It is therefore important to minimize positional errors during the path execution phase.

In cases where wheel-slip occurs, open-loop estimation methods usually fail. However, using the method described in this paper, an accurate estimate of position can be maintained even when kinematic constraints are violated.

In most movement schemes, dead-reckoning errors are an accepted part of the movement sequence. This is unfortunate, because in many cases dead-reckoning proves to be inaccurate. If a more accurate method were available, fewer intermediate localizations would be needed. This would in turn free computational resources that could be used on higher level tasks.

Other researchers have implemented similar dead-reckoning correction techniques. In [1] a towed robot (called a trailer) is used, which has highly accurate wheel encoders and a rotary encoder on the connection link to determine the relative direction of the trailer. This method has proven to be accurate, reducing the dead-reckoning errors by an order of magnitude or more. However, the added bulk of the trailer can complicate the motion of the robot making it difficult to navigate in close quarters, especially when moving backwards. [2] used extra wheel encoder in addition to the driving wheel encoders. The redundant sensor information are fused using fuzzy algorithm. [3] introduced a dead-reckoning system fusing a magnetic compass, rate-gyroscope, encoders and four ultrasonic range sensors based on the extended Kalman filter. Experimental results using the encoder and the sonar sensors for localization is described in [4].

The process model for the extended Kalman filter, which is widely used for the mobile robot localization, is investigated thoroughly in [5].

Another method of correcting dead-reckoning errors in navigation uses optical flow. In [6], optical flow was used to aid in navigation of an omnidirectional robot. A CCD camera was positioned at a 45° downward angle to the ground in front of the robot. The optical flow obtained was then combined with the results of dead-reckoning using the maximum likelihood technique. The method used to calculate optical flow is quite complex, requiring a large number of computations to give good results.

GPS has been used to correct position estimations by adjusting kinematic parameters. In [7] GPS was used to correct for heading and step size in a pedestrian navigation system. When GPS is available, the parameters are adjusted, so that when GPS service is unavailable, still a good estimate of position is maintained. This method is easily portable to

mobile robots. However, GPS accuracy is limited, so when fine positional control is necessary it can prove ineffective. GPS is further limited by the fact that it will only work in outdoor environments where line-of-sight to at least 3 satellites is possible. For outdoor mobile robot application, [8] used the extended Kalman filter for the GPS and encoder data. In order to obtain better accuracy of estimation, an arbitrator is designed to switch between the extended Kalman filter and dead reckoning. [9] also fused ABS sensor information and GPS signal using the extended Kalman filter.

[10] developed inertial navigation methods for mobile robots. A series of solid state gyros, accelerometers and tilt sensors were employed in an extended Kalman filter to estimate position and orientation. Using accelerometers to determine position and orientation, however, has several drawbacks such as 1-8 cm/s drift rate. Also, the minimum detectable acceleration can sometimes be too large to detect small motion. Inertial methods can be quite computationally intensive, expensive and complex.

II. Optical Sensor Interpretation and Analysis

Commercial optical flow sensors are used in this paper. The sensors chosen are those currently used in optical mice. The optical flow calculation is done by the built-in digital signal processor.

Different from the conventional optical flow analysis, the sensor provides only dx and dy but not $d\theta$ because $d\theta$ information is not necessary for Windows interface. Thus, the location of the sensors is important because $d\theta$ is not available directly from the sensor. The change in orientation is clearly important when updating the robot position. Therefore, any errors in dx or dy will greatly affect the accuracy of the rigid-body method. An additional limitation of the proposed system is a nominal 2.5mm height requirement of the optical sensor. This limits use of the method as implemented to flat smooth surfaces. Our navigation experiments (section IV) are done on concrete surface, which is quite flat. However, the sensor misses significant amount of movement on carpeted surface due to height variation. This issue is discussed in section III.

A. Single Optical Sensor

Optical flow information taken directly from an optical sensor will not be useful unless the sensor is positioned at the point of interest on the robot. However, because the optical sensor only provides displacements in the x and y directions, information about the angular displacement of the robot is lost. Another method must therefore be used to determine the movement of the robot.

The robot can be viewed as a rigid-body where the velocity at the sensor (Δx/Δt and Δy/Δt) is known. The kinematic constraints of a differential drive robot allow the calculation of movement, given this velocity. Specifically, the center of the robot is assumed to move only in a direction perpendicular to the wheel axis. Using this information, the rigid-body velocity equation is

$$\mathbf{V}_r = \mathbf{V}_s + \mathbf{\Omega}_r \times \mathbf{r}_{s/r} \tag{1}$$

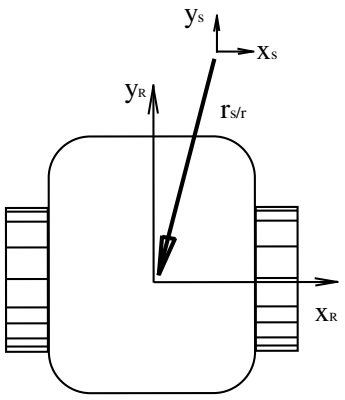

Fig. 1. Rigid-Body Model for Sensor Interpretation

Where $\mathbf{V}_r$ is the velocity of the center of the robot, $\mathbf{V}_s = \begin{bmatrix} V_x & V_y & 0 \end{bmatrix}^T$ is the sensor velocity, $\mathbf{\Omega}_r = \begin{bmatrix} 0 & 0 & \omega_r \end{bmatrix}^T$ is the angular velocity of the robot, and $\mathbf{r}_{s/r} = \begin{bmatrix} r_x & r_y & 0 \end{bmatrix}^T$ is the vector from the location of the sensor to the robot center (Fig. 1) As long as there is no x-direction wheel slip, the velocity of the robot at the center will always be in the y-direction. From the rigid-body velocity equation and the kinematic constraints, we obtain the following two equations.

$$\omega_r = -\frac{V_x}{r_y} \tag{2}$$

$$V_r = V_y + \frac{r_x}{r_y} V_x \tag{3}$$

The robot's velocity can now be easily translated to the global coordinate system.

B. Multiple Optical Sensors

Looking again at the rigid-body model, two sensors at different locations give ample information to determine the motion of the rigid-body without any additional kinematic constraints. In this case the rigid-body model leads to the following four equations.

$$V_{r,x} = V_{1,x} + \omega_r \cdot r_{1,y} \tag{4}$$
$$V_{r,y} = V_{1,y} - \omega_r \cdot r_{1,x} \tag{5}$$
$$V_{r,x} = V_{2,x} + \omega_r \cdot r_{2,y} \tag{6}$$
$$V_{r,y} = V_{2,y} - \omega_r \cdot r_{2,x} \tag{7}$$

where $V_{r,x}$ and $V_{r,y}$ are the velocities of the center of the robot, ω_r is the angular velocity of the robot, $V_{i,x}$ and $V_{i,y}$ are the i-th sensor velocities and $r_{i,x}$ and $r_{i,y}$ are the x and y distances from the sensor positions to the robot center.

To solve this system of equations for the angular and linear velocities of the robot, the system is written as follows:

$$\begin{bmatrix} 1 & 0 & -r_{1,y} \\ 0 & 1 & r_{1,x} \\ 1 & 0 & -r_{2,y} \\ 0 & 1 & r_{2,x} \end{bmatrix} \cdot \begin{bmatrix} V_{r,x} \\ V_{r,y} \\ \omega_r \end{bmatrix} = \begin{bmatrix} V_{1,x} \\ V_{1,y} \\ V_{2,x} \\ V_{2,y} \end{bmatrix} \tag{8}$$

The least-squares method can now be used to solve this equation, resulting in the following:

$$\begin{bmatrix} V_{r,x} \\ V_{r,y} \\ \omega_r \end{bmatrix} = \{ \begin{bmatrix} 1 & 0 & -r_{1,y} \\ 0 & 1 & r_{1,x} \\ 1 & 0 & -r_{2,y} \\ 0 & 1 & r_{2,x} \end{bmatrix}^T \begin{bmatrix} 1 & 0 & -r_{1,y} \\ 0 & 1 & r_{1,x} \\ 1 & 0 & -r_{2,y} \\ 0 & 1 & r_{2,x} \end{bmatrix} \}^{-1} \begin{bmatrix} 1 & 0 & -r_{1,y} \\ 0 & 1 & r_{1,x} \\ 1 & 0 & -r_{2,y} \\ 0 & 1 & r_{2,x} \end{bmatrix}^T \begin{bmatrix} V_{1,x} \\ V_{1,y} \\ V_{2,x} \\ V_{2,y} \end{bmatrix} \quad (9)$$

Linear and angular velocities, ($V_{r,x}$, $V_{r,y}$, ω_r) from Eq. 9 are with respect to the robot's local coordinates. Position and orientation of the robot in global coordinates are derived from the following equations.

$$\theta_{robot} = \int (\omega_r) dt \quad (10)$$

$$X_{robot} = \int (V_{r,x} \cos\theta_{robot} - V_{r,y} \sin\theta_{robot}) dt \quad (11)$$

$$Y_{robot} = \int (V_{r,x} \sin\theta_{robot} + V_{r,y} \cos\theta_{robot}) dt \quad (12)$$

Although multiple sensors do not increase accuracy when compared to a single sensor, the kinematic requirements can now be removed. This gives a powerful method to determine intermediate estimates of the robot position and orientation without the need for any kinematic constraints.

III. Estimation using Extended Kalman Filter

Instead of relying on the optical flow sensor(s) only (Eqs. 2, 3 or Eq. 9) , we can use the encoder data in order to get more accurate estimation, even though the dead-reckoning problem exists. One of the commonly used estimation using multiple measurements is based on the extended Kalman filter. Especially, the optical flow sensor requires quite strict requirement of the vertical distance from the surface to be 2.5mm. Even with the specially designed sensor bracket with pre-tensioned springs in order to keep the nominal distance as required, the distance varies depending on the surface condition. Thus, the optical flow sensor does not detect the movement if the distance variation is large. Therefore, relying on only the optical flow sensor does not provide robust estimation. In this section, we describe the model and procedure of estimating the robot configuration from the extended Kalman filter.

The plant model of the mobile robot is generally described as the following equation.

$$\mathbf{x}(k+1) = \Phi[\mathbf{x}(k), \mathbf{u}(k)] + \mathbf{v}(k) \quad (13)$$

where $\Phi()$ is the state transition function and $\mathbf{v}()$ is zero-mean Gaussian noise with covariance $\mathbf{C}_v(k)$. State vector, $\mathbf{x}$ at time k is defined as

$$\mathbf{x}(k) = \begin{bmatrix} V_x(k) & V_y(k) & \omega(k) \end{bmatrix}^T \quad (14)$$

The configuration of the mobile robot is estimated by integrating the state vector.

$$\begin{bmatrix} x(t) \\ y(t) \\ \theta(t) \end{bmatrix} = \begin{bmatrix} \int_0^t V_x dt \\ \int_0^t V_y dt \\ \int_0^t \omega dt \end{bmatrix} \quad (15)$$

For this robot the control input $\mathbf{u}(k)$ can be described as

$$\mathbf{u}(k) = \begin{bmatrix} V_R(k) & V_L(k) \end{bmatrix}^T \quad (16)$$

where $V_R(k)$ and $V_L(k)$ are right and left wheel velocities, respectively. The the nonlinear plant model is then given by

$$\Phi[\mathbf{x}(k), \mathbf{u}(k)] = \begin{bmatrix} V_x(k) + \frac{V_R(k)+V_L(k)}{2} \cos\theta(k) \\ V_y(k) + \frac{V_R(k)+V_L(k)}{2} \sin\theta(k) \\ \omega(k) + \frac{V_R(k)-V_L(k)}{D} \end{bmatrix} \quad (17)$$

The measurement model is as follows:

$$\mathbf{z}_i(k) = \mathbf{h}[\mathbf{x}(k), \xi] + \mathbf{w}_i(k) \quad (18)$$

where $\mathbf{z}_i$ is a vector of measurements of sensor i

$$\mathbf{z}_i(k) = \begin{bmatrix} v_{i,x}(k) & v_{i,y}(k) \end{bmatrix}^T \quad (19)$$

and $\mathbf{w}_i(k)$ is zero-mean Gaussian noise with covariance $\mathbf{C}_w(k)$. Note that each sensor measures only v_x and v_y but not angular velocity. Using the optical flow sensors, the measurement model is linear and this takes the form

$$\mathbf{z}_i(k) = \mathbf{\Lambda}_E \mathbf{x}(k) + \mathbf{w}_i(k) \quad (20)$$

and an estimate of the state can be recovered from the measurements as

$$\hat{\mathbf{x}}(k) = \mathbf{\Lambda}_E^{-1} \mathbf{z}_i(k) \quad (21)$$

if it is assumed $\mathbf{\Lambda}_E$ is invertible. In previous section, Kinematic relation between optical flow sensor measurement and mobile robot velocity are shown and in the measurement model,

$$\begin{bmatrix} v_{i,x}(k) \\ v_{i,y}(k) \end{bmatrix} = \begin{bmatrix} V_x(k) - \omega(k) r_{i,y} \\ V_y(k) + \omega(k) r_{i,x} \end{bmatrix} + \mathbf{w}_i(k) \quad (22)$$

by rearranging the equation,

$$\mathbf{z}_i(k) = \begin{bmatrix} 1 & 0 & -r_{i,y} \\ 0 & 1 & r_{i,x} \end{bmatrix} \begin{bmatrix} V_x(k) \\ V_y(k) \\ \omega(k) \end{bmatrix} + \mathbf{w}_i(k) \quad (23)$$

hence,

$$\mathbf{\Lambda}_E = \begin{bmatrix} 1 & 0 & -r_{i,y} \\ 0 & 1 & r_{i,x} \end{bmatrix} \quad (24)$$

and $\mathbf{\Lambda}_E$ is not square. As long as there's no kinematic violation (for instance, no side slip), pseudo inverse is used as

$$\hat{\mathbf{x}}(k) = \mathbf{\Lambda}_E^{\#} \mathbf{z}_i(k) \quad (25)$$

where,

$$\mathbf{\Lambda}_E^{\#} = \mathbf{\Lambda}_E^T \left(\mathbf{\Lambda}_E \mathbf{\Lambda}_E^T \right)^{-1} \quad (26)$$

we update the uncertainty of the state as

$$\mathbf{P}(k+1|\,k) = \mathbf{\Phi}(k)\mathbf{P}(k)\mathbf{\Phi}^T(k) + \mathbf{C}_v(k) \tag{27}$$

The Kalman gain can be computed as

$$\mathbf{K}(k+1) = \mathbf{P}(k+1|\,k)\mathbf{\Lambda}_E^T \left[\mathbf{\Lambda}_E(k+1)\mathbf{P}(k+1|\,k)\mathbf{\Lambda}_E^T(k+1) + \mathbf{C}_w(k+1)\right]^{-1} \tag{28}$$

The difference between the actual sensor data and the predicted one using the state estimate is

$$\mathbf{r}(k+1) = \mathbf{z}(k+1) - \mathbf{h}\left[\hat{\mathbf{x}}(k+1|\,k), \xi\right] \tag{29}$$

The revised state estimate is then given by

$$\hat{\mathbf{x}}(k+1) = \hat{\mathbf{x}}(k+1|\,k) + \mathbf{K}(k+1)\mathbf{r}(k+1) \tag{30}$$

and the revised state covariance matrix is given by

$$\mathbf{P}(k+1) = \left[\mathbf{I} - \mathbf{K}(k+1)\mathbf{\Lambda}_E\right]\mathbf{P}(k+1|\,k) \tag{31}$$

In order to check the performance of the extended Kalman filter, large amount of random numbers are added to the optical flow sensor readings intentionally. Using a single optical flow sensor, the estimation is drifted and the final estimated position is far from the actual measured position. The robot trajectory is smooth, and dead-reckoning results are better than the one estimated using an optical flow sensor. The extended Kalman filter based estimation combines both the encoder and the optical flow sensor measurements and shows the best result. However, when there exists side-slip, the plant model is not valid during the slip and the extended Kalman filter does not provide useful estimation.

IV. Experimental Verification

A. Sensor Specifications

The optical sensor used is the Agilent ADNS-2051. This CMOS chip allows fast, accurate, optical sensing of microscopic images. The sensor is capable of 800 counts per inch while moving at up to 14 inches per second. Successive images are used to calculate the Δx and Δy values at up to 2300 frames per second. A microcontroller is used which communicates directly with the ADNS-2051. This chip communicates with the ADNS-2051, setting modes and communicating with the PC or other device. The optical sensor uses successive images to interpret the movement between images. Images are taken of a point on the ground directly underneath the robot. The images are 16 by 16 pixels and represent a microscopic area.

B. Robot Specifications

The mobile robot includes a 12 inch, round plastic base, two 1 amp 12 Volt DC geared motors, two castor wheels, two 6 inch diameter rubber wheels, and two HEDS-5500 encoders attached to the drive wheel axles. The differential driven wheels allow turns to be made in place. This symmetric design offers high maneuverability. The maximum straight line speed is approximately 78 feet per minute (50 revolutions per minute at the drive wheels).

C. Robot Controller

The controller consists of 3 devices, the robot controller, the sensor controller, and a PC The PC is a "remote brain" in our setup. The PC communicates with both controllers using two separate serial communication ports. A command is sent from the PC to the robot controller specifying the speed desired for each wheel. The robot controller follows this command for each wheel. The sensor controller sends data about the movement of the robot back to the PC. The PC interprets the data from the sensor controller. The PC then corrects for any discrepancies in the observed position of the robot and the desired position.

D. Robot Kinematics

The forward kinematics for a differential drive mobile robot are based on velocity equations of the left and right wheels. Fig. 3 shows the robot model used to find the kinematics of

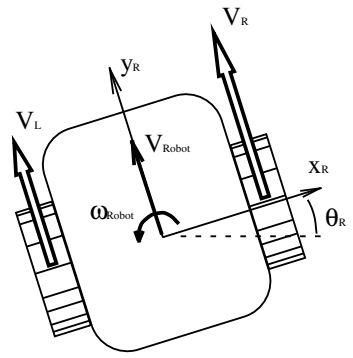

Fig. 2. Robot Coordinates for Forward Kinematic Model

the system. The equations obtained from this model are:

$$\dot{x} = \frac{V_L + V_R}{2}\cos\theta \tag{32}$$

$$\dot{y} = \frac{V_L + V_R}{2}\sin\theta \tag{33}$$

$$\dot{\theta} = \frac{V_R - V_L}{D} \tag{34}$$

Where V_L and V_R are the left and right wheel velocities and D is the distance between the wheels. Using these equations, the forward kinematics solution is easily obtained in a step-wise fashion. This method is also known as dead-reckoning and is a common solution to the intermediate estimation problem. This model is a benchmark for comparisons with the presented method.

E. Intermediate Position Estimation

Experiments were performed comparing the intermediate positional estimates using dead-reckoning and the multiple sensor rigid-body method. In experiments where kinematic constraints were upheld ($V_{robot,x} = 0$ and no wheel slip) as in Fig. 4, little performance difference can be seen between the methods. When the kinematic constraints were intentionally violated with disturbance as in Fig. 5, however, only the

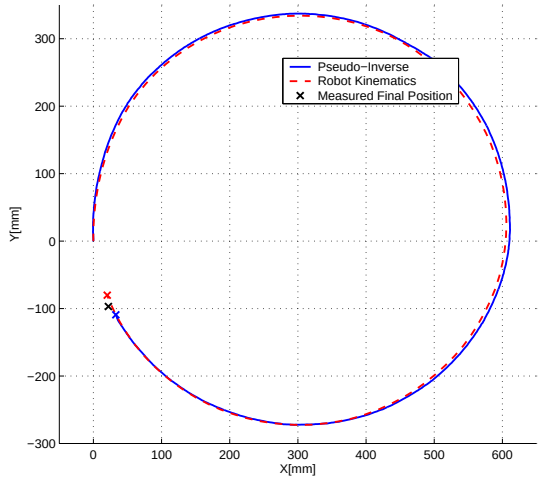

Fig. 3. Localization Results with Kinematics Upheld

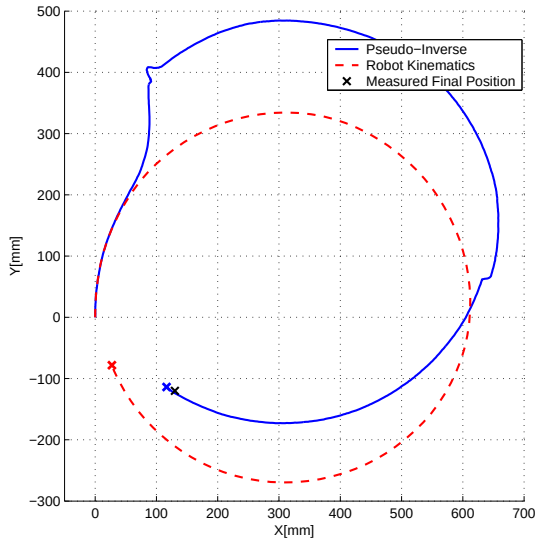

Fig. 4. Localization Results with Kinematic Violation

multiple sensor method gave good results. Using single optical flow sensor, the robot configuration is estimated correctly even with wheel slip because the robot kinematics is still valid. From the following condition, we can detect the existence of the wheel slip.

$$\|\mathbf{X}_{enc} - \mathbf{X}_{single}\| \geq \varepsilon_1 \tag{35}$$

where $\mathbf{X}_{enc}$ is the robot's position in Cartesian coordinate estimated from purely encoder readings, $\mathbf{X}_{single}$ is from one optical flow sensor and ε_1 is a positive constant. Similarly, we can detect the kinematic violation such as side-slip from the following condition

$$\|\mathbf{X}_{single} - \mathbf{X}_{dual}\| \geq \varepsilon_2 \tag{36}$$

where $\mathbf{X}_{dual}$ is the robot's position using two optical flow sensors based on Pseudo inverse and ε_2 is a positive constant.

F. Potential Field based Navigation

In order to test the effectiveness of the intermediate estimation method described, navigation tests were performed. The multiple sensor rigid-body models and dead-reckoning were compared under various conditions.

A potential field based approach was used to generate velocity commands for the mobile robot to follow. In this approach, the distance to the goal point and the presence of obstacles affects the planned motion of the robot. An online method was used where the desired V_{robot} and ω_{robot} were generated depending upon the current robot location/orientation and the location of the goal point. Obstacles were assumed not to be present, but can easily be added.

The resulting $V_{desired}$ and $\omega_{desired}$ were then translated to $\dot{x}_{desired}$, $\dot{y}_{desired}$ and $\dot{\theta}_{desired}$ which were in turn translated to wheel velocities, V_L and V_R, using the inverse kinematic relationship.

In tests where no wheel slip or kinematic violations occurred both methods were similarly accurate. Where the rigid-body method is most beneficial is in the presence of kinematic violations, such as wheel slip. Fig. 6 shows an actual test run where wheel slip was forced. As one can see, the robot has a sudden change of direction at the beginning of movement. The multiple-sensor rigid-body method is able to detect this movement, which allows the potential field based navigation method to adjust its course accordingly. Even when kinematic constraints are blatantly violated, as in Fig. 7 where the robot was pushed off course, the multiple sensor rigid-body method still detects the proper change in position and orientation so that the robot successfully reaches the goal.

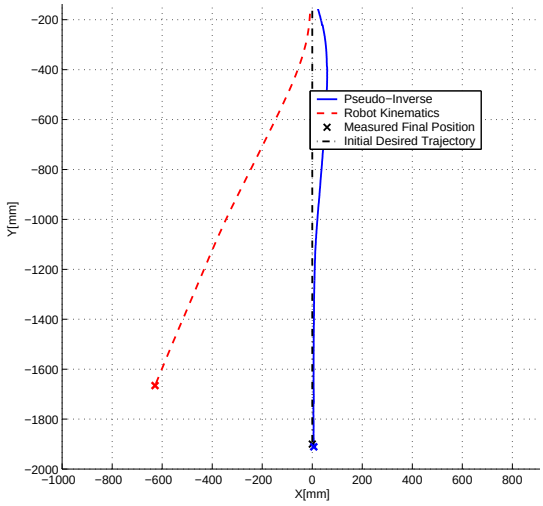

Fig. 5. Navigation with Wheel Slip

V. Conclusion

Using optical flow sensors, we have derived a method which can accurately give intermediate position estimates. The location of the sensors has been investigated in order to minimize potential errors introduced by incorrect sensor

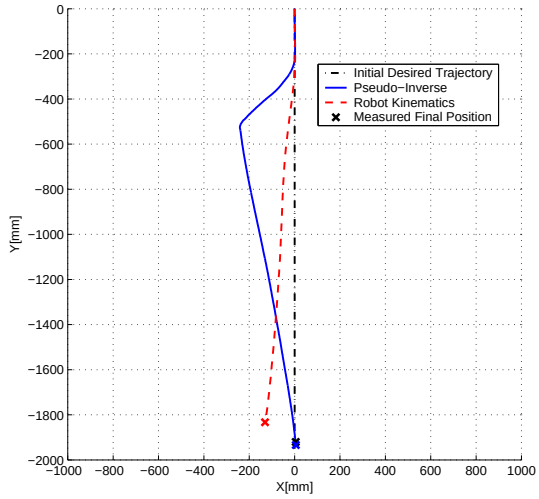

Fig. 6. Navigation with Kinematic Violation

readings. In order to use both encoder and optical flow sensor, the extended Kalman filter is designed for optimal estimation. When there exist incorrect optical flow sensor readings, the extended Kalman filter based estimation gives better estimates. However, with kinematic violation such as side slip, the filter doesn't work well because it is based on the kinematic model. Experimentally, the multiple sensor rigid-body method has proven to be more accurate and more robust than dead-reckoning. Both systematic and non-systematic errors can be detected and a good estimate of location and orientation be maintained.

VI. Acknowledgments

This research is supported by grant No. R01-2003-000-10336-0 from the Basic Research Program of the Korea Science & Engineering Foundation

References

[1] Johann Borenstein, "Internal Correction of Dead-reckoning Errors With the Smart Encoder Trailer," *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems '94*, vol. 1, pp. 127–134, Sep. 1994.

[2] De Xu, Min Tan, Gang Chen, "An Improved Dead Reckoning Method for Mobile Robot with Redundant Odometry Information" *Proc. of the 7th Int. Conf. on Control, Automation, Robotics and Vision*, pp. 631–636, Dec. 2002.

[3] Ching-Chih Tsai, "A Localization System of a Mobile Robot by Fusing Dead-Reckoning and Ultrasonic Measurements" *IEEE Transactions on Instrumentation and Measurement*, vol. 47 no. 5, pp. 1399–1404, Oct. 1998.

[4] Leopoldo Jetto, Sauro Longhi, Giuseppe Venturini, "Development and Experimental Validation of an Adaptive Extended Kalman Filter for the Localization of Mobile Robots" *IEEE Transactions on Robotics and Automation*, vol. 15 no. 2, pp. 219–229, Feb. 2003.

[5] Simon J. Julier, Hugh F. Durrant-Whyte, "On the Role of Process Models in Autonomous Land Vehicle Navigation Systems" *IEEE Transactions on Robotics and Automation*, vol. 19 no. 1, pp. 1–14, Feb. 2003.

[6] K. Nagatani, S. Tachibana, M. Sofne, Y. Tanaka, "Improvement of Odometry for Omnidirectional Vehicle using Optical Flow Information," *Proc. 2000 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, vol. 1, pp. 468–473, Oct. 2000.

[7] R. Jirawimut, P. Ptasinski, V. Garaj, F. Cecelja, W. Balachandran, "A Method for Dead Reckoning Parameter Correction in Pedestrian Navigation System," *Proc. of the 18th IEEE Instrumentation and Measurement Technology Conference*, vol. 3, pp. 1554–1558, May 2001.

[8] Bo He, Danwei Wang, Minhtuan Pham, Tieniu Yu, "GPS/Encoder based Precise navigation for a 4WS Mobile Robot" *Proc. of the 7th Int. Conf. on Control, Automation, Robotics and Vision*, pp. 1256–1261, Dec. 2002.

[9] Philippe Bonnifait, Pascal Bouron, Paul Crubille, Dominique Meizel, "Data Fusion of Four ABS Sensors and GPS for an Enhanced Localization of Car-like Vehicles" *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 1597–1602, May 2001.

[10] Billur Barshan and Hugh F. Durant-Whyte, "Inertial Navigation Systems for Mobile Robots," *IEEE Transactions on Robotics and Automation*, vol. 11 issue 3, pp. 328–342, Jun. 1995.