Seventh International Conference on Control, Automation, Robotics And Vision (ICARCV'02), Dec 2002, Singapore

GPS/Encoder Based Precise Navigation for a 4WS Mobile Robot

Bo He, Danwei Wang, Minhtuan Pham, Tieniu Yu
School of Electrical and Electronics Engineering
Nanyang Technological University, Singapore 639798, {ebhe, edwwang}@ntu.edu.sg

Abstract

In this paper, position and orientation estimation with high accuracy based on GPS and encoders for a a four-wheel-steering vehicle (4WS) mobile robot is addressed. An architecture of position and orientation estimation is proposed, which consists of two Extended Kalman Filters and a processing unit of Runga-Kutta-based dead reckoning. The first EKF fuses data from six encoders to estimate the velocity of vehicle and sideslip angle. The second EKF is applied to the estimation of position and orientation based on the measurement from precise GPS data and output from first EKF. To obtain better accuracy of estimation, an arbitrator is designed to switch between EKF2 and dead reckoning. The results and analysis of experiments are presented to show the effectiveness of the proposed approach.

1 Introduction

Precise navigation is being researched and applied in various fields, especially for outdoor mobile robots [1,2,3], such as cargo container transportation at sea port, precision agriculture, military vehicles, and etc. As we know, applied sensors and relevant data processing primarily decide the accuracy of navigation. In our research, GPS and encoders are used as navigation sensors. As an absolute positioning sensor, GPS has been commercially applied in vehicles localization and navigation as well as military areas. Since June 2000, the SA (Selective Availability error) degradation has been turned off; Standard GPS precision of 10-meters-level enhances civil commercial applications, especially for general navigation tasks. A GPS receiver working in Real-Time Kinematics mode can offer centimeter-level accuracy for our vehicle. In addition to GPS, the vehicle is equipped with four incremental encoders and two absolute encoders to provide information of wheels for estimation of velocity and sideslip angle of vehicle. To estimate position and orientation on the basis of GPS data and encoder data, a novel architecture of position and orientation estimate is proposed, which consists of two Extended Kalman Filters and a processing unit of Runga-Kutta based dead reckoning. The first EKF fuses data from six encoders to estimate the velocity and sideslip angle at a reference point of vehicle. The second EKF estimates position and orientation based on the measurement from GPS data and output from the first EKF. Further, to obtain the better accuracy of estimation, a dedicated arbitrator is designed to switch on or off appropriate processing unit (EKF2 or dead reckoning). The results and analysis of experiments are presented to show the effectiveness of the proposed approach.

This paper is organized as follows. Section 2 describes our research platform, a 4WS vehicle, and its kinematics model. Section 3 studies position and orientation estimation in details, including the architecture of data fusion, estimation algorithms, and arbitration. Then, results and analysis of some experiments are provided in section 4. Finally section 5 presents conclusions.

2 Kinematics Model of a Mobile Robot

Our research platform, Cycab, is a computer-aid and DC-motor-driven four-wheel-steering vehicle. The kinematic model used in this paper is based on the one in [1] that was derived for a 4WS vehicle. The motion of the vehicle can be described in the global coordinates by using the "bicycle model"[1,4] as illustrated in Fig.1. The reference point CG is chosen at the center of gravity of the vehicle. Point O is the Instant Center of Turn. δ_f and δ_r are the steering angles of front and rear wheels. (x, y, ψ) defines the Configuration of the vehicle body, where, (x, y) is the position of the reference point in the global coordinate and ψ is the vehicle heading angle.

The kinematic model can be derived based on principles of geometry as follows

$$\begin{cases} x(k+1) = x(k) + \Delta.v\cos[\psi(k)+\beta] \\ y(k+1) = y(k) + \Delta.v\sin[\psi(k)+\beta] \\ \psi(k+1) = \psi(k) + \Delta.\dfrac{v}{l_r.\cos\delta_r}\sin(\beta-\delta_r) \end{cases} \tag{1}$$

where, Δ is the sampling interval.

3 Position and Orientation Estimate with High Accuracy

Position and orientation estimation aims to produce the vehicle configuration, (x, y, ψ), using the measurement data from GPS and encoders mounted on the steering and driving systems.

3.1 Architecture

When reception is good, GPS measurements provide the vehicle position (x, y), but not the vehicle heading angle. However, GPS signals are not available at all time; thus navigation requires more than the GPS. Encoders are a common type of sensors in the industry. They are used to record the speed or relative position of a rotating shaft. To estimate the configuration of a vehicle, encoders are not reliable because of one significant flaw of encoder itself - drift.

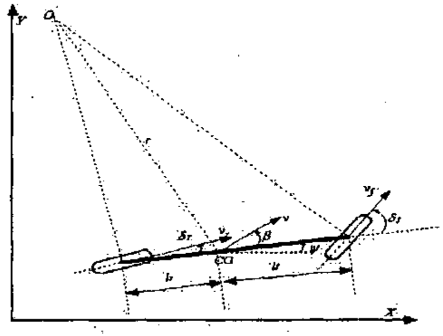


Fig.1 Kinematics Model

For any relative navigation sensors, drift is a general flaw, which will get larger through the time due to error accumulation. As such, combination of the absolute sensors e.g. GPS or compass and relative sensors e.g. encoder, gyroscope and accelerometer is a natural solution [5,6]. In this paper, we propose an architecture to estimate the configuration of the vehicle based on Extended Kalman Filtering and Runga-Kutta method as illustrated in Fig.2. In this architecture, two Extended Kalman Filters are deployed; one (EKF1) is used to estimate the velocity and sideslip angle of the vehicle by the measurement of wheels velocity and steering angle. Another one (EKF2) is used to fuse absolute positioning information from GPS and the output of the first Extended Kalman Filter to estimate the configuration of the vehicle, i.e. (x, y, ψ). A block of R-K is referred to as dead reckoning based on Runga-Kutta method. An Arbitrator and its relative two virtual switches in this architecture, decide which of estimate solutions is on service. The details of the above mentioned blocks of algorithms would be presented in the sequel.

Wheel velocities v_{fl}, v_{fr}, v_{rl} and v_{rr}, are obtained from incremental encoders mounted at each of four wheels. δ_f and δ_r are steering angles of two front wheels and two rear wheels respectively, which are measured by two absolute encoders installed at the hydraulic jacks on the vehicle.

3.2 Estimation Algorithms

In EKF1, the state vector **X** is $[v, \beta, \rho]^T$, ρ is the curvature at center of gravity of the vehicle. Assuming that the velocity and sideslip angle of the vehicle keep constant within one sampling interval, the process model is as follows:

$$\begin{cases} v(k+1) = v(k) + w_v(k) \\ \beta(k+1) = \beta(k) + w_\beta(k) \\ \rho(k+1) = \rho(k) + w_\rho(k) \end{cases} \tag{1}$$

$[w_v, w_\beta, w_\rho]^T$ is process error vector, a zero-mean white noise with Q1 covariance matrix. The above assumption is valid while the vehicle is driven smoothly without sharp accelerating/decelerating or turning.

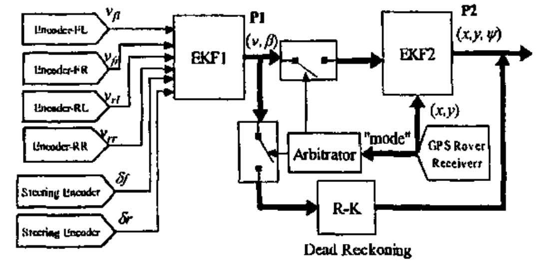


Fig.2 Architecture of position and orientation estimation

Measurement model should be the equations that express relationships between four wheels velocity, two steering angles and state vector $[v, \beta, \rho]^T$. Fig.3 represents the kinematics of vehicle based on four-wheels model. r is the instant radius at point of CG, i.e. the reciprocal of ρ. The velocities at the four wheels of the vehicle are conformed as equation (2).

$$\vec{v}_{fl} = v\cos\beta \begin{bmatrix} 1 - \dfrac{b\rho}{\cos\beta} \\ \tan\delta_f \end{bmatrix} \qquad \vec{v}_{fr} = v\cos\beta \begin{bmatrix} 1 + \dfrac{b\rho}{\cos\beta} \\ \tan\delta_f \end{bmatrix} \tag{2}$$

$$\vec{v}_{rl} = v\cos\beta \begin{bmatrix} 1 - \dfrac{b\rho}{\cos\beta} \\ \tan\delta_r \end{bmatrix} \qquad \vec{v}_{rr} = v\cos\beta \begin{bmatrix} 1 + \dfrac{b\rho}{\cos\beta} \\ \tan\delta_r \end{bmatrix}$$

Taking the magnitudes only:

$$v_{fl} = v\cos\beta\sqrt{\left(1-\frac{b\rho}{\cos\beta}\right)^2+\tan^2\delta_f} \quad v_{fr} = v\cos\beta\sqrt{\left(1+\frac{b\rho}{\cos\beta}\right)^2+\tan^2\delta_f} \tag{3}$$

$$v_{rl} = v\cos\beta\sqrt{\left(1-\frac{b\rho}{\cos\beta}\right)^2+\tan^2\delta_r} \quad v_{rr} = v\cos\beta\sqrt{\left(1+\frac{b\rho}{\cos\beta}\right)^2+\tan^2\delta_r}$$

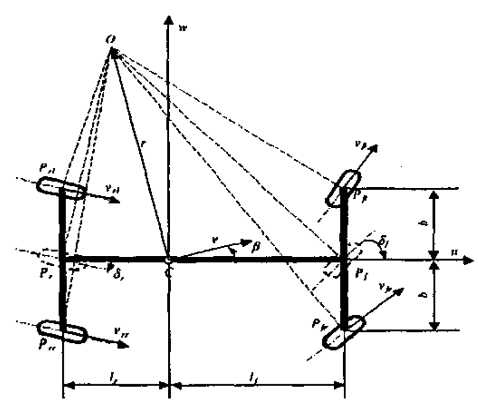


Fig.3 Kinematics model in vehicular coordinate

The front and rear steering angles δ_f and δ_r can be derived by analyzing the Fig.3 as follows.

$$\tan\frac{\delta_f}{2} = \frac{-\cos\beta + \sqrt{l_f^2\rho^2 + 2l_f\rho\sin\beta + 1}}{\rho l_f + \sin\beta}$$

$$\tan\frac{\delta_r}{2} = \frac{\cos\beta - \sqrt{l_r^2\rho^2 - 2l_r\rho\sin\beta + 1}}{\rho l_r - \sin\beta} \tag{4}$$

Combining equations (3) and (4) yields a discretized measurement model as equations (5) and (6).

$$\mathbf{z}(k) = \mathbf{h1}[\mathbf{x}(k), k] + \mathbf{v}(k) \tag{5}$$

$$\mathbf{h1}[\mathbf{x}(k), k] = \begin{bmatrix} v(k)\cos\beta(k)\sqrt{(1-\frac{b\rho(k)}{\cos\beta(k)})^2 + [\tan\delta_f(k)]^2} \\ v(k)\cos\beta(k)\sqrt{(1+\frac{b\rho(k)}{\cos\beta(k)})^2 + [\tan\delta_f(k)]^2} \\ v(k)\cos\beta(k)\sqrt{(1-\frac{b\rho(k)}{\cos\beta(k)})^2 + [\tan\delta_r(k)]^2} \\ v(k)\cos\beta(k)\sqrt{(1+\frac{b\rho(k)}{\cos\beta(k)})^2 + [\tan\delta_r(k)]^2} \\ \frac{-\cos\beta(k) + \sqrt{l_f^2\rho(k)^2 + 2l_f\rho(k)\sin\beta(k) + 1}}{\rho(k)l_f + \sin\beta(k)} \\ \frac{\cos\beta(k) - \sqrt{l_r^2\rho(k)^2 - 2l_r\rho(k)\sin\beta(k) + 1}}{\rho(k)l_r - \sin\beta(k)} \end{bmatrix} \tag{6}$$

where, $\mathbf{Z}=[v_{fl}, v_{fr}, v_{rl}, v_{rr}, \tan(\delta_f/2), \tan(\delta_r/2)]^T$ is the measurement vector; $\mathbf{v}$ is the measurement error vector, which is a zero-mean white noise vector with $\mathbf{R1}$ covariance matrix. $\mathbf{h1}[\mathbf{x}(k),k]$ in equation (6) is a nonlinear transition function.

To apply Extended Kalman Filtering, the Jacobian matrices with respect to state vector should be worked out [7]. The Jacobian matrix of transition function of process model is an identity matrix, the Jacobian $\mathbf{H1}$ ("1" here means the index of EKF1) of $\mathbf{h1}$ of measurement model is as presented by the following equation.

$$\mathbf{H1}[\hat{\mathbf{x}}(k+1) \mid k), k] = \left.\frac{\partial \mathbf{h1}(x,k)}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}(k+1|k)} \tag{7}$$

Details can be found in Appendix.

EKF1 works in the following steps:

Step1: State Prediction and Measurement Prediction

$\hat{\mathbf{x}}(k+1 \mid k) = \hat{\mathbf{x}}(k \mid k)$

$\mathbf{P1}(k+1 \mid k) = \hat{\mathbf{x}}(k \mid k)\mathbf{P1}(k \mid k)\hat{\mathbf{x}}(k \mid k) + \mathbf{Q1}(k)$

$\hat{\mathbf{Z}}(k+1 \mid k) = \mathbf{h1}[\hat{\mathbf{X}}(k+1 \mid k), k]$

$\mathbf{S}(k+1 \mid k) = \mathbf{H1}[\hat{\mathbf{X}}(k+1 \mid k), k]\mathbf{P1}(k+1 \mid k)\mathbf{H1}^T[\hat{\mathbf{X}}(k+1 \mid k), k] + \mathbf{R1}(k)$

Step 2: Compute EKF1 Gain

$\mathbf{K}(k+1 \mid k) = \mathbf{P1}(k+1 \mid k)\mathbf{H1}^T[\hat{\mathbf{X}}(k+1 \mid k), k]\mathbf{S}(k+1)^{-1}$

Step 3: Update Estimate

$\hat{\mathbf{X}}(k+1 \mid k+1) = \hat{\mathbf{X}}(k+1 \mid k) + \mathbf{K}(k+1)[\mathbf{Z}(k+1) - \hat{\mathbf{Z}}(k+1 \mid k)]$

$\mathbf{P1}(k+1 \mid k+1) = \mathbf{P1}(k+1 \mid k) - \mathbf{K}(k+1)\mathbf{S}(k+1)\mathbf{K}^T(k+1)$

Repeat the above process for $k = 0, 1, \ldots$

In EKF2, the state vector is $[x, y, \psi]^T$, and the estimation results of EKF1 are taken as the control input here. The estimate error covariance $\mathbf{P1}$ of EKF1 is used to decide the initial error covariance of the process model.

The process model of EKF2 is:

$$\mathbf{x}(k+1) = \mathbf{f2}[\mathbf{x}(k), k] + \mathbf{w}(k) \tag{8}$$

where, $\mathbf{f2}[\mathbf{x}(k),k]$ is of the form of equation (1), $\mathbf{w}$ is the error vector of process model with zero-mean and Q2 the error covariance.

Equations (9) and (10) are the measurement mode of the EKF2. The measurement vector $\mathbf{Z}$ is directly recorded data from GPS positioning information with zero-mean and R2 error covariance.

$$\mathbf{z}(k) = \mathbf{h2}[\mathbf{x}(k), k] + \mathbf{v}(k) \tag{9}$$

$$\begin{cases} h_1(k) = x_1(k) = x(k) \\ h_2(k) = x_2(k) = y(k) \end{cases} \tag{10}$$

Jacobian matrix $\mathbf{F2}$ of transition function vector $\mathbf{f2}$ with respect to state vector and Jacobian matrix $\mathbf{H2}$ of transition function vector $\mathbf{h2}$ with respect to state vector are presented in the following forms respectively:

$$\mathbf{F2} = \left.\frac{\partial \mathbf{f2}(\mathbf{x},k)}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}(k|k)} = \begin{bmatrix} 1 & 0 & -v\sin[\psi(k)+\beta(k)] \\ 0 & 1 & v\cos[\psi(k)+\beta(k)] \\ 0 & 0 & 1 \end{bmatrix} \tag{11}$$

$$\mathbf{H2}[\hat{\mathbf{x}}(k+1) \mid k), k] = \left.\frac{\partial \mathbf{h2}(x,k)}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}(k+1|k)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{12}$$

EKF2 works in the following steps:

Step1: State Prediction and Measurement Prediction

$\hat{\mathbf{x}}(k+1 \mid k) = \mathbf{f2}[\hat{\mathbf{x}}(k \mid k), k]$

$\mathbf{P2}(k+1 \mid k) = \mathbf{F2}[\hat{\mathbf{x}}(k \mid k), k]\mathbf{P2}(k \mid k)\mathbf{F2}^T[\hat{\mathbf{x}}(k \mid k), k] + \mathbf{Q1}(k)$

$\hat{\mathbf{Z}}(k+1 \mid k) = \mathbf{h2}[\hat{\mathbf{X}}(k+1 \mid k), k]$

$\mathbf{S}(k+1 \mid k) = \mathbf{H2}[\hat{\mathbf{X}}(k+1 \mid k), k]\mathbf{P2}(k+1 \mid k)\mathbf{H2}^T[\hat{\mathbf{X}}(k+1 \mid k), k] + \mathbf{R2}(k)$

Step 2: Compute EKF2 Gain

$\mathbf{K}(k+1 \mid k) = \mathbf{P2}(k+1 \mid k)\mathbf{H1}^T[\hat{\mathbf{X}}(k+1 \mid k), k]\mathbf{S}(k+1)^{-1}$

Step 3: Update Estimate

$\hat{\mathbf{X}}(k+1 \mid k+1) = \hat{\mathbf{X}}(k+1 \mid k) + \mathbf{K}(k+1)[\mathbf{Z}(k+1) - \hat{\mathbf{Z}}(k+1 \mid k)]$

$\mathbf{P2}(k+1 \mid k+1) = \mathbf{P2}(k+1 \mid k) - \mathbf{K}(k+1)\mathbf{S}(k+1)\mathbf{K}^T(k+1)$

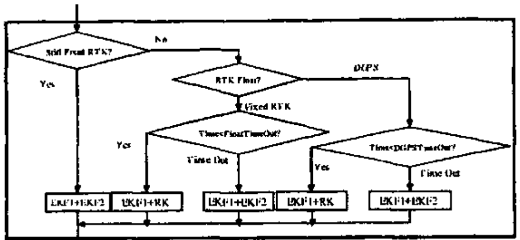
Repeat the above process for $k = 0, 1, \ldots$

By fusing the data from encoders and GPS, the position and orientation of the vehicle can be estimated using the above process. The estimation can achieve high accuracy due to the high accurate data from Fixed RTK of GPS. However, in practice, Fixed RTK mode is not constantly available during the vehicle driving. There are many cases that could result in the satellites signals mask or deterioration, for examples, the vehicle passing by a bridge or high building, or being close to a high tree, etc. Even at a nearly open-view site, the deterioration of GPS signals could occur due to ionosphere activities or satellites failure. In such cases, GPS operation mode would be degraded from Fixed RTK to RTK Float or even DGPS (We would not cover the situations of completely losing GPS signals or degrading to standard GPS in this paper), the estimation precision also will deteriorate significantly as well. To compensate the lost precision as much as possible, a dead reckoning is applied to replace EKF2 to produce the position and orientation of the vehicle during the above deterioration stages. The dead reckoning here is a Runga-Kutta-based algorithm to solve the differential equation (8).

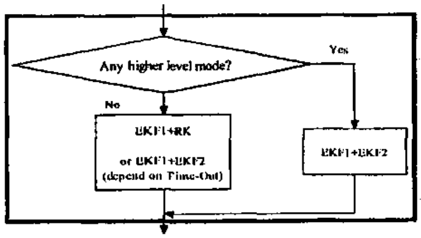
3.3 Arbitration Mechanism

To choose the best estimate outputs from EKF2 or the dead-reckoning, an Arbitrator (see Fig.4) is designed to decide which estimate is online. In the NMEA data output of GPS rover receiver there is a message segment of "GPS quality indictor", which indicts the current GPS modes by digits. The arbitrator utilizes such messages to switch the algorithms between EKF2 and Runga-Kutta-based dead reckoning. From Fig.4 we notice that, the dead reckoning has to subject to a timeout checking, its "lifespan" is limited in a preset interval. Or else, the drift of encoders data through the time could go beyond a specified error bound, even beyond the error bound of DGPS or RTK Float.

Another issue is that the dead reckoning must be initialized by the latest results of estimation. In another word, the dead reckoning should be calibrated by the latest available GPS data before it is on service.



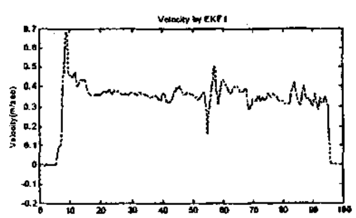
(a) Arbitration when vehicle runs in Fixed RTK mode



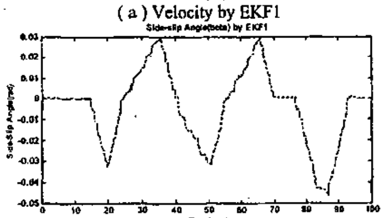
(b) Arbitration when vehicle runs in non-RTK(Fixed) mode
Fig.4 Arbitration mechanism

4 Experimental Results

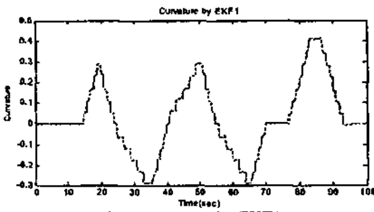
Experiments are carried out in a car park. There are some low trees and houses nearby. In most cases, the rover GPS receiver on our vehicle works in Fixed RTK mode. Estimation outputs of EKF1 are shown on the Fig.5, estimation can quickly converge even initializing the state vector with some bias. To evaluate the estimation, a quasi-true velocity by post-processing of GPS data is used for comparison; the square error shown in Fig.5(d) reflects the performance of estimation to some extent.
Fig.6 and Fig.7 show the trajectories of two experiments. They basically represent the cases that most possibly occur during the vehicle is running at a nearly open-view site.



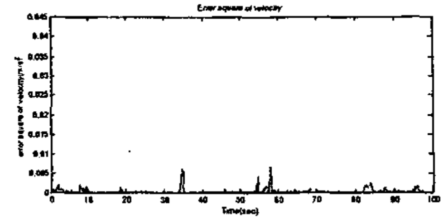
(a) Velocity by EKF1



(b) Sideslip angle by EKF1



(c) Curvature by EKF1



(d) Square error of velocity
Fig.5 Results of EKF1

In the experiment 1, the vehicle starts running from the right top of Fig.6 with GPS working in the Fixed RTK. The dotted line is the trajectory recorded by GPS data, the solid line is produced by the estimator represented in Fig.2. The estimation mostly coincides with the one by GPS, because EKF1 + EKF2 is being used to estimate the states according to the arbitration mechanism. From starting point, GPS is in the mode of Fixed RTK, good results are produced by EKF1+EKF2. When the vehicle reaches at segment AB, the DGPS mode occurs and a significant error (magnified profile is shown in the right window) emerges accordingly. Thus RTK1+RK algorithm is carried out within the Timeout period (15 seconds for DGPS, 20 seconds for RTK Float). The upper little arrow A points to the epoch entering DGPS, the lower arrow B points to the epoch recovering to Fixed RTK. Clearly the estimation of EKF1+RK is quite good in such case. In the last period (after point B), GPS works in Fixed RTK again, the estimator applies EKF1+EKF2 accordingly.

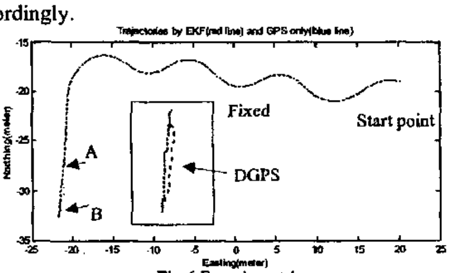


Fig.6 Experiment 1

Experiment 2 presents a worse situation of GPS signal (see Fig.7). A signal block occurs by covering the antenna of the rover receiver intentionally for a short term. This experiment aims to show the performance of the estimator while the GPS works in bad condition. The vehicle starts running with GPS working in Fixed RTK mode, EKF1+EKF2 is used to estimate the states. From point A, GPS signal is lost suddenly, there is no normal position data output (in such case, zeros are taken as the position output, so the dotted line jumps to the origin), GPS receiver is in the "blind" status. At the point B, GPS signal recovers, and enter the DGPS mode first, then enter the Float RTK mode at point C, finally restores the high precise mode, i.e. Fixed RTK at point D. From point B to point D, a typical complete initialization procedure of RTK is performed, which takes about 30 seconds.

From the point A, the EKF1+RK (dead reckoning) is switched on to estimate the configuration of the vehicle. In this case, the dead reckoning only lasts 15 seconds. under the rule of Arbitrator. After the Timeout, the estimator has to switches to EKF1+EKF2 at around the point C due to the drift of the encoders.

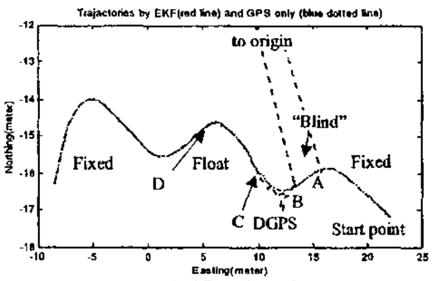


Fig.7 Experiment 2

5 Conclusions

Real time kinematics positioning of GPS provides very high precision (centimeter-level) for the navigation of vehicles or outdoor mobile robots. But practically it has to be used together with other sensors if taking into consideration of the deterioration of GPS signals. In this paper, a 4WS vehicle is considered and we propose an estimator which consists of two extended Kalman Filters, Runga-Kutta-based dead reckoning unit and an Arbitrator, to estimate the position and orientation of the vehicle. The results of the experiments show the efficiency of estimation with high accuracy, especially for the cases of short-term deterioration of satellites signals.

References

[1] D. Wang, E.K. Tay and M. Zribi, "Modeling of an AGV for handling heavy containers", *3 rd International Conference on Advanced Mechatronics*, Japan, pp.67-72, August 1998.

[2] Michael Lee O'Connor. "Carrier-Phase Differential GPS for Automatic Control of Land Vehicles". PhD thesis. Stanford Univ.,1997.

[3] M. Iida, M. Kudou, K. Ono, M. Umeda. "Automatic Following Control for Agricultural Vehicle". *AMC*, Nagoya, pp.158-162, 2000.

[4] E. K. Tay. "Modeling, Analysis and Control Design for An Automated Guided Vehicle". MS thesis, Nanyang Tech. Univ., 1999

[5] K. Kobayashi, K. Watanabe. "Accurate Navigation via Sensor Fusion of Differential GPS and Rate-Gyro". *IMTC*, Hamamatsu, pp.556-559, May 1994.

[6] Eric Abbott, David Powell. "Land-Vehicle Navigation Using GPS". *Proceeding of IEEE*, vol.87 no.1, pp.145-162, Jan. 1999.

[7] J. A. Farrell, M. Barth. "The Global Positioning System and Inertial Navigation". McGraw-Hill. 1999.

Appendix

Supposing,

$$A_f=\sqrt{l_f^{\,2}\rho^2+2l_f\rho\sin\beta+1}\ ;B_f=\frac{-\cos\beta+\sqrt{l_f^{\,2}\rho^2+2l_f\rho\sin\beta+1}}{l_f\rho+\sin\beta}$$

$$A_r=\sqrt{l_r^{\,2}\rho^2-2l_r\rho\sin\beta+1}\ ;B_r=\frac{\cos\beta-\sqrt{l_r^{\,2}\rho^2-2l_r\rho\sin\beta+1}}{l_r\rho-\sin\beta}$$

The Jacobian Matrix of the EKF1 is

$$\mathbf{H1}[\hat{\mathbf{x}}(k+1)\,|\,k),k]=\left.\frac{\partial \mathbf{h1}(x,k)}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}(k+1|k)}$$

$$=\left.\begin{bmatrix}\partial v_{fl}/\partial v & \partial v_{fl}/\partial\beta & \partial v_{fl}/\partial\rho\\ \partial v_{fr}/\partial v & \partial v_{fr}/\partial\beta & \partial v_{fr}/\partial\rho\\ \partial v_{rl}/\partial v & \partial v_{rl}/\partial\beta & \partial v_{rl}/\partial\rho\\ \partial v_{rr}/\partial v & \partial v_{rr}/\partial\beta & \partial v_{rr}/\partial\rho\\ \partial\tan(\delta_f/2)/\partial v & \partial\tan(\delta_f/2)/\partial\beta & \partial\tan(\delta_f/2)/\partial\rho\\ \partial\tan(\delta_r/2)/\partial v & \partial\tan(\delta_r/2)/\partial\beta & \partial\tan(\delta_r/2)/\partial\rho\end{bmatrix}\right|_k$$

Where,

$$\partial v_{fl}/\partial v=\cos\beta\sqrt{\left(1-\frac{b\rho}{\cos\beta}\right)^2+\frac{4B_f^{\,2}}{(1-B_f^{\,2})^2}}$$

$$\partial v_{fl}/\partial\beta=\frac{1}{2}v\cos\beta\frac{\left\{\begin{array}{l}2\dfrac{(1+\frac{b\rho}{\cos\beta})b}{\cos\beta}-\dfrac{8B_rl_r}{(1-B_r^{\,2})^2A_r}-\dfrac{8B_r^{\,2}l_r}{(l_r\rho-\sin\beta)(1-B_r^{\,2})^2}\\+\dfrac{-16B_r^{\,3}l_r(l_r\rho-\sin\beta)-16B_r^{\,4}l_rA_r}{(1-B_r^{\,2})(l_r\rho-\sin\beta)A_r}\end{array}\right\}}{\sqrt{\left(1+\frac{b\rho}{\cos\beta}\right)^2+\frac{4B_r^{\,2}}{(1-B_r^{\,2})^2}}}$$

$$\partial v_{fl}/\partial\rho=\frac{1}{2}v\cos\beta\frac{\left\{\begin{array}{l}-2\dfrac{(1-\frac{b\rho}{\cos\beta})b}{\cos\beta}+\dfrac{8B_fl_f}{(1-B_f^{\,2})^2A_f}-\dfrac{8B_f^{\,2}l_f}{(l_f\rho+\sin\beta)(1-B_f^{\,2})^2}\\+\dfrac{16B_f^{\,3}l_f(l_f\rho+\sin\beta)-16B_f^{\,4}l_fA_f}{(1-B_f^{\,2})(l_f\rho+\sin\beta)A_f}\end{array}\right\}}{\sqrt{\left(1-\frac{b\rho}{\cos\beta}\right)^2+\frac{4B_f^{\,2}}{(1-B_f^{\,2})^2}}}$$

$$\partial v_{fr}/\partial v=\cos\beta\sqrt{\left(1+\frac{b\rho}{\cos\beta}\right)^2+\frac{4B_f^{\,2}}{(1-B_f^{\,2})^2}}$$

$$\partial v_{fr} \Big/ \partial \beta = -v \sin\beta \sqrt{\left(1+\frac{b\rho}{\cos\beta}\right)^2 + \frac{4B_f^{\,2}}{(1-B_f^{\,2})^2}}$$

$$+\frac{1}{2} v \cos\beta \frac{\left\{\begin{array}{l} 2\dfrac{\left(1+\dfrac{b\rho}{\cos\beta}\right)b\rho\sin\beta}{\cos^2\beta} + \dfrac{8B_f\left(\sin\beta+\dfrac{l_f\rho\cos\beta}{A_f}\right)-8B_f^{\,2}\cos\beta}{(l_f\rho+\sin\beta)(1-B_f^{\,2})^2} \\ +\dfrac{16B_f^{\,3}\left(\sin\beta+\dfrac{l_f\rho\cos\beta}{A_f}-B_f\cos\beta\right)}{(l_f\rho+\sin\beta)(1-B_f^{\,2})} \end{array}\right\}}{\sqrt{\left(1+\dfrac{b\rho}{\cos\beta}\right)^2+\dfrac{4B_f^{\,2}}{(1-B_f^{\,2})^2}}}$$

$$\partial v_{fr} \Big/ \partial \rho - \frac{1}{2} v \cos\beta \frac{\left\{\begin{array}{l} 2\dfrac{\left(1+\dfrac{b\rho}{\cos\beta}\right)b}{\cos\beta} + \dfrac{8B_f l_f}{(1-B_f^{\,2})^2 A_f} - \dfrac{8B_f^{\,2} l_f}{(l_f\rho+\sin\beta)(1-B_f^{\,2})^2} \\ +\dfrac{16B_f^{\,3} l_f (l_f\rho+\sin\beta)+16B_f^{\,4} l_f A_f}{(1-B_f^{\,2})(l_f\rho+\sin\beta)A_f} \end{array}\right\}}{\sqrt{\left(1+\dfrac{b\rho}{\cos\beta}\right)^2+\dfrac{4B_f^{\,2}}{(1-B_f^{\,2})^2}}}$$

$$\partial v_{rl} \Big/ \partial v = \cos\beta \sqrt{\left(1-\frac{b\rho}{\cos\beta}\right)^2 + \frac{4B_r^{\,2}}{(1-B_r^{\,2})^2}}$$

$$\partial v_{rl} \Big/ \partial \beta = -v \sin\beta \sqrt{\left(1-\frac{b\rho}{\cos\beta}\right)^2 + \frac{4B_r^{\,2}}{(1-B_r^{\,2})^2}}$$

$$+\frac{1}{2} v \cos\beta \frac{\left\{\begin{array}{l} -2\dfrac{\left(1-\dfrac{b\rho}{\cos\beta}\right)b\rho\sin\beta}{\cos^2\beta} + \dfrac{8B_r\left(-\sin\beta+\dfrac{l_r\rho\cos\beta}{A_r}\right)+8B_r^{\,2}\cos\beta}{(l_r\rho-\sin\beta)(1-B_r^{\,2})^2} \\ +\dfrac{16B_r^{\,3}\left(-\sin\beta+\dfrac{l_r\rho\cos\beta}{A_r}+B_r\cos\beta\right)}{(l_r\rho-\sin\beta)(1-B_r^{\,2})} \end{array}\right\}}{\sqrt{\left(1-\dfrac{b\rho}{\cos\beta}\right)^2+\dfrac{4B_r^{\,2}}{(1-B_r^{\,2})^2}}}$$

$$\partial v_{rl} \Big/ \partial \rho = \frac{1}{2} v \cos\beta \frac{\left\{\begin{array}{l} -2\dfrac{\left(1-\dfrac{b\rho}{\cos\beta}\right)b}{\cos\beta} - \dfrac{8B_r l_r}{(1-B_r^{\,2})^2 A_r} - \dfrac{8B_r^{\,2} l_r}{(l_r\rho-\sin\beta)(1-B_r^{\,2})^2} \\ +\dfrac{-16B_r^{\,3} l_r (l_r\rho+\sin\beta)-16B_r^{\,4} l_r A_r}{(1-B_r^{\,2})(l_r\rho-\sin\beta)A_r} \end{array}\right\}}{\sqrt{\left(1-\dfrac{b\rho}{\cos\beta}\right)^3+\dfrac{4B_r^{\,2}}{(1-B_r^{\,2})^3}}}$$

$$\partial v_{rr} \Big/ \partial v = \cos\beta \sqrt{\left(1+\frac{b\rho}{\cos\beta}\right)^2 + \frac{4B_r^{\,2}}{(1-B_r^{\,2})^2}}$$

$$\partial v_{rr} \Big/ \partial \beta = -v \sin\beta \sqrt{\left(1+\frac{b\rho}{\cos\beta}\right)^2 + \frac{4B_r^{\,2}}{(1-B_r^{\,2})^2}}$$

$$+\frac{1}{2} v \cos\beta \frac{\left\{\begin{array}{l} 2\dfrac{\left(1+\dfrac{b\rho}{\cos\beta}\right)b\rho\sin\beta}{\cos^2\beta} + \dfrac{8B_r\left(-\sin\beta+\dfrac{l_r\rho\cos\beta}{A_r}\right)+8B_r^{\,2}\cos\beta}{(l_r\rho-\sin\beta)(1-B_r^{\,3})^2} \\ +\dfrac{16B_r^{\,3}\left(-\sin\beta+\dfrac{l_r\rho\cos\beta}{A_r}+B_r\cos\beta\right)}{(l_r\rho-\sin\beta)(1-B_r^{\,2})} \end{array}\right\}}{\sqrt{\left(1+\dfrac{b\rho}{\cos\beta}\right)^2+\dfrac{4B_r^{\,2}}{(1-B_r^{\,2})^2}}}$$

$$\partial v_{rr} \Big/ \partial \rho = \frac{1}{2} v \cos\beta \frac{\left\{\begin{array}{l} 2\dfrac{\left(1+\dfrac{b\rho}{\cos\beta}\right)b}{\cos\beta} - \dfrac{8B_r l_r}{(1-B_r^{\,3})^2 A_r} \\ -\dfrac{8B_r^{\,2} l_r}{(l_r\rho-\sin\beta)(1-B_r^{\,3})^2} + \dfrac{-16B_r^{\,3} l_r (l_r\rho-\sin\beta)-16B_r^{\,4} l_r A_r}{(1-B_r^{\,3})(l_r\rho-\sin\beta)A_r} \end{array}\right\}}{\sqrt{\left(1+\dfrac{b\rho}{\cos\beta}\right)^2+\dfrac{4B_r^{\,3}}{(1-B_r^{\,2})^2}}}$$

$$\partial \tan(\delta_f/2) \Big/ \partial v = 0$$

$$\partial \tan(\delta_f/2) \Big/ \partial \beta = \frac{\sin\beta + \dfrac{l_f\rho\cos\beta}{A_f}}{l_f\rho+\sin\beta} + \frac{B_f\cos\beta}{l_f\rho+\sin\beta}$$

$$\partial \tan(\delta_f/2) \Big/ \partial \rho = \frac{l_f}{A_f} - \frac{B_f l_f}{l_f\rho+\sin\beta}$$

$$\partial \tan(\delta_r/2) \Big/ \partial v = 0$$

$$\partial \tan(\delta_r/2) \Big/ \partial \beta = \frac{-\sin\beta + \dfrac{l_r\rho\cos\beta}{A_r}}{l_r\rho-\sin\beta} + \frac{B_r\cos\beta}{l_r\rho-\sin\beta}$$

$$\partial \tan(\delta_r/2) \Big/ \partial \rho = \frac{l_r}{A_r} - \frac{B_r l_r}{l_r\rho-\sin\beta}$$