Chapter 9

GPS Aided Encoder-Based Dead-Reckoning

Encoder-based navigation is a form of odometry or dead-reckoning for land vehicles. The basic physical setup is illustrated in Figure 9.1. On an axle of length L, two wheels are equipped with encoders that measure discrete changes in wheel rotation. The two wheels are separately actuated so that each can rotate independently on the axle. The goal is to use the encoder measurements with GPS aiding to maintain an estimate of the tangent plane position $\mathbf{p}^t = [n, e, h]^\top$ of the center of the axle.

The general three-dimensional navigation problem will be discussed briefly in Section 9.7. In the main body of this chapter, we consider a vehicle maneuvering on a planar surface. We assume that the plane is horizontal with respect to the Earth surface at a point $\mathbf{p}_0$ that will serve as the origin of the tangent plane navigation frame. The height of the vehicle is known to be h_r; therefore, $h = h_r$ and the navigation position estimation problem is two dimensional. The horizontal position $\mathbf{p} = [n, e]^\top$ and the tangent frame velocity $\mathbf{v}^t = [\dot{n}, \dot{e}]^\top$ vectors have only two components, and the pitch and roll angles are identically zero. The only attitude variable is the vehicle yaw angle ψ. The kinematic state dimension is three: n, e, and ψ. The kinematic inputs are the (change of) encoder pulse counts $\Delta e_L(k)$ and $\Delta e_R(k)$ for encoders attached to the left and right axles, respectively, during a time interval dT. The vector of calibration coefficients contains the wheel radii $[R_R, R_L]^\top$. Thus the state vector is $\mathbf{x} = [n, e, \psi, R_R, R_L]^\top$.

Due to the horizontal planar surface assumption the altitude h is known. This approach easily extends to non-horizontal planar surfaces and can be extended to smooth non-planar surfaces with roll and pitch being input

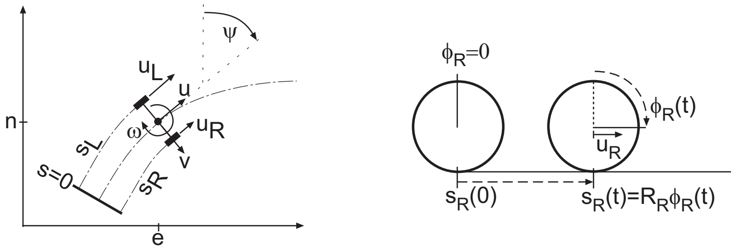

Figure 9.1: Variable definitions for encoder-based navigation. In the left image, the center of the axle has linear velocity $\mathbf{v} = [u, v]^\top$ and angular rate ω about the vehicle vertical axis. The vehicle yaw angle relative to north is ψ. The velocity and arc length of the left wheel are denoted by u_L and s_L, respectively. The velocity and arc length of the right wheel are denoted by u_R and s_R, respectively. The image on the right shows the relationships between the angular rotation ϕ_R of the right wheel and the variables u_R and s_R.

variables determined by additional sensors. For accuracy of the estimate, an important issue is that the wheels do not slip.

The outline for this chapter is as follows. Section 9.1 presents a brief discussion of encoders with the encoder model that will be used in this chapter. The vehicle kinematic model is presented in Section 9.2. Section 9.3 presents both continuous-time and discrete-time navigation mechanization equations. The continuous-time equations are useful for derivation of the error state dynamic equations. The discrete-time equations are suitable for navigation implementation. The error state dynamic model is derived in Section 9.4. Two alternative GPS measurement prediction and residual measurement model equations are presented and compared in Section 9.5. The system observability is considered in Section 9.6. Finally, Section 9.7 considers an extension of the method presented in this chapter to remove the planar surface assumption.

9.1 Encoder Model

Encoders are often implemented as a thin disk attached to the axle with narrow slits cut radially through the disk. A light source and light detector are placed on opposite sides of the disk. As the axle and disk rotate, the light detector senses light pulses. Using multiple light detectors and arranging the slits at different distances from the disk center, it is also possible to sense the direction of rotation. Various types of encoders exist, but the common performance factors are that the encoder output is an integer proportional to the angular rotation of the axle. In the case where

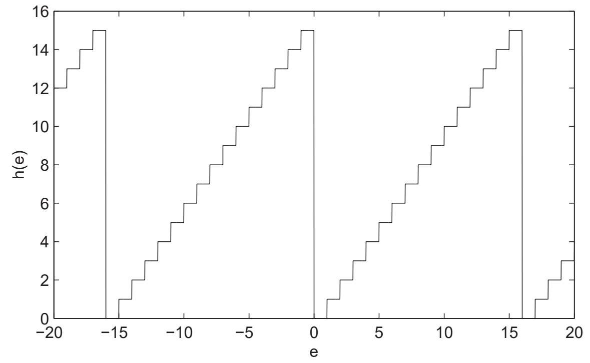

Figure 9.2: Example absolute encoder output with scale factor $C = 16$ counts per revolution.

incremental encoders are used, the design would need to include either interrupt driven or hardware accumulators of the increments to utilize the methods described herein. An important issue is to ensure that no encoder increments are missed.

Although the encoder outputs are discrete in nature, the wheel angular rotations are continuous. Because the continuous-time and continuous state representation is convenient for development of the error models, this section will model continuous valued encoder signals e_L and e_R. The encoder sensor output signals will be quantized versions of the continuous signal $\tilde{e}_L$ and $\tilde{e}_R$. The continuous signal will be used for the development of the error model equations while only the quantized sensor output can be used in the implemented navigation equations.

Let C denote the encoder scale factor in units of counts per revolution. The variables (ϕ_L, ϕ_R) denote the radian angular rotation about the axle of the left and right wheels, respectively. Then, the shaft angle and the (continuous) encoder signal are related by

$$\left.\begin{aligned} e_L(t) &= \tfrac{C}{2\pi}\phi_L(t) \\ e_R(t) &= \tfrac{C}{2\pi}\phi_R(t) \end{aligned}\right\} \tag{9.1}$$

where the signals e_L and e_R are real-valued. The encoder sensor outputs are quantized to integer values and reflected back to always lie in the range $[0, C-1]$:

$$\left.\begin{aligned} \tilde{e}_L &= h(e_L) \\ \tilde{e}_R &= h(e_R). \end{aligned}\right\} \tag{9.2}$$

A typical form of the nonlinearity h is shown in Figure 9.2 for a scale factor of $C = 16$ counts per revolution. This value of C is selected for clarity

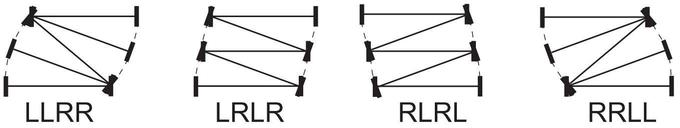

Figure 9.3: Net vehicle position changes resulting from various sequences of two left and right rotational increments.

of the figure only. Typical values of C are much larger. Note that C is a perfectly known constant value.

The algorithm for processing the encoder outputs must run fast enough to not miss increments and to maintain the order in which the increments occur. Figure 9.3 illustrates four possible scenarios involving two left increments indicated by an 'L' and four right increments indicated by an 'R'. The only differences between the scenarios are the ordering of occurrence of the incremental wheel rotations. Note that, as shown in Figure 9.3, a LRLR sequence of increments is distinct from a RRLL sequence. While both result in the vehicle having the same yaw angle, the first moves the vehicle slightly to the right, while the second moves the vehicle slightly to the left. The algorithm must also be designed to correctly accommodate the overflow that occurs at integer multiples of C counts.

9.2 Kinematic Model

Assuming that there is no wheel slip and that the lateral vehicle velocity is zero, the vehicle kinematics are described by

$$\left.\begin{aligned} \dot{\mathbf{p}} &= \begin{bmatrix} \dot{n} \\ \dot{e} \end{bmatrix} = \begin{bmatrix} \cos(\psi) \\ \sin(\psi) \end{bmatrix} u \\ \dot{\psi} &= \omega \end{aligned}\right\} \tag{9.3}$$

where ω is the body frame yaw rate and $\mathbf{v}^b = [u, v]^\top$ is the body frame velocity vector. The body and tangent frame velocity vectors are related by $\mathbf{v}^t = \mathbf{R}_b^t \mathbf{v}^b$, where

$$\mathbf{R}_b^t = \begin{bmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & \cos(\psi) \end{bmatrix}.$$

By the zero lateral velocity assumption, the lateral velocity $v = 0$.

Using the kinematic relationship $\mathbf{v}_w = \mathbf{v}^b + \boldsymbol{\omega}_{tb}^b \times \mathbf{R}$ where the superscript w denotes wheel and the subscript b denotes body, the body frame linear velocity of the center of each wheel is

$$\left.\begin{aligned} u_L &= u + \tfrac{L}{2}\omega \\ u_R &= u - \tfrac{L}{2}\omega. \end{aligned}\right\} \tag{9.4}$$

These linear velocities are illustrated in Figure 9.1. The function in eqn. (9.4) from (u, ω) to (u_L, u_R) can be inverted so that the speed and angular rate can be computed when the wheel velocities are known:

$$\left.\begin{aligned} u &= \tfrac{1}{2}\left(u_L + u_R\right) \\ \omega &= \tfrac{1}{L}\left(u_L - u_R\right). \end{aligned}\right\} \tag{9.5}$$

Let (s_L, s_R) denote the arc length traveled by the left and right wheels, respectively. Due to the no slip assumption and assuming that $s_L(0) = s_R(0) = \phi_L(0) = \phi_R(0) = 0$, we have the conditions

$$\left.\begin{aligned} s_L &= R_L \phi_L & \qquad \phi_L &= \tfrac{1}{R_L} s_L \\ s_R &= R_R \phi_R & \qquad \phi_R &= \tfrac{1}{R_R} s_R \end{aligned}\right\} \tag{9.6}$$

where the angular rotation of the left and right axles (ϕ_L, ϕ_R) are defined in eqn. (9.1). By the definition of arc length, we have

$$\left.\begin{aligned} \dot{s}_L &= u_L \\ \dot{s}_R &= u_R. \end{aligned}\right\} \tag{9.7}$$

Therefore, by substituting eqn. (9.7) into the derivative of eqn. (9.6) and solving for (u_L, u_R) we obtain

$$\left.\begin{aligned} u_L &= R_L \dot{\phi}_L \\ u_R &= R_R \dot{\phi}_R. \end{aligned}\right\} \tag{9.8}$$

Finally, combining eqns. (9.1), (9.3), (9.5) and (9.8) we obtain

$$\left.\begin{aligned} \dot{n} &= \tfrac{1}{2}\tfrac{2\pi}{C}\left(R_L \dot{e}_L + R_R \dot{e}_R\right)\cos(\psi) + \omega_n \\ \dot{e} &= \tfrac{1}{2}\tfrac{2\pi}{C}\left(R_L \dot{e}_L + R_R \dot{e}_R\right)\sin(\psi) + \omega_e \\ \dot{\psi} &= \tfrac{1}{L}\tfrac{2\pi}{C}\left(R_L \dot{e}_L - R_R \dot{e}_R\right) + \omega_\psi. \end{aligned}\right\} \tag{9.9}$$

The signals ω_n, ω_e, and ω_ψ represent errors due to, for example, violation of the no-slip assumption. These errors are not necessarily stationary and mutually uncorrelated random signals; however, unless some form of slip detection algorithm is incorporated, ω_n, ω_e, and ω_ψ would typically be modeled as independent random processes.

The length of the wheel radii R_L and R_R vary relative to their nominal values. The variation depends on the type of tire as well as various unpredictable factors. The wheel radii are modeled as

$$R_L = R_0 + \delta R_L \tag{9.10}$$

$$R_R = R_0 + \delta R_R \tag{9.11}$$

where R_0 is the known nominal value of the wheel radii. The error in the wheel radii will be modeled as scalar Gauss-Markov processes

$$\begin{aligned} \delta\dot{R}_L &= -\lambda_R \delta R_L + \omega_L && (9.12) \\ \delta\dot{R}_R &= -\lambda_R \delta R_R + \omega_R && (9.13) \end{aligned}$$

with $\lambda_R > 0$, $E\langle \delta R_L(0)\rangle = E\langle \delta R_R(0)\rangle = 0$, and

$$var(\delta R_R(0)) = var(\delta R_L(0)) = \bar{P}_R > 0.$$

The correlation time $\frac{1}{\lambda_R}$ is selected large (i.e., a correlation time of 10 hours), because the wheel radii are very slowly time varying. The signals ω_L and ω_R are each white, Gaussian noise processes with PSD of $\sigma_R^2 = 2\lambda_R \bar{P}_R$. This value of σ_R^2 is selected so that the steady state solution of eqn. (4.100) when applied to eqns. (9.12-9.13) is $\bar{P}_R$.

9.3 Encoder Navigation Equations

This section presents the navigation mechanization equations in both continuous and discrete-time. The continuous-time equations will be used to develop the navigation error equations. The discrete-time equations will be used for implementation of the dead-reckoning navigation system.

9.3.1 Continuous-Time: Theory

Based on augmented state vector

$$\hat{\mathbf{x}} = [\hat{n}, \hat{e}, \hat{\psi}, \hat{R}_l, \hat{R}_r]^\top, \tag{9.14}$$

the continuous-time navigation system mechanization equations are

$$\left.\begin{aligned} \dot{\hat{n}} &= \tfrac{1}{2}\tfrac{2\pi}{C}\left(\hat{R}_L \dot{e}_L + \hat{R}_R \dot{e}_R\right)\cos(\hat{\psi}) \\ \dot{\hat{e}} &= \tfrac{1}{2}\tfrac{2\pi}{C}\left(\hat{R}_L \dot{e}_L + \hat{R}_R \dot{e}_R\right)\sin(\hat{\psi}) \\ \dot{\hat{\psi}} &= \tfrac{1}{L}\tfrac{2\pi}{C}\left(\hat{R}_L \dot{e}_L - \hat{R}_R \dot{e}_R\right) \end{aligned}\right\} \tag{9.15}$$

with

$$\left.\begin{aligned} \hat{R}_R &= R_0 + \delta\hat{R}_R \\ \hat{R}_L &= R_0 + \delta\hat{R}_L \end{aligned}\right\} \tag{9.16}$$

where $\delta\hat{R}_L(0)$ and $\delta\hat{R}_R(0)$ will be estimated by the Kalman filter. Eqns. (9.15) and (9.16) will be used in Section 9.4 to derive the state space error model.

9.3.2 Discrete-Time: Implementation

This section considers the integration of eqns. (9.15) over a time increment ΔT. Depending on the type of encoder used, the increment ΔT can be realized by different means. For a pair of incremental encoders, the update would ideally be performed for each encoder increment. In this case, the time increment would not be constant, but would be determined by the vehicle motion. Alternatively, the cumulative encoder increments could be read at a fixed rate. The main criteria in the selection of that sample rate is that the accumulated encoder increments must each be small and none can be missed. The encoder increments will be denoted as

$$\begin{aligned} \Delta e_L(k) &= e_L(t_k) - e_L(t_{k-1}) \\ \Delta e_R(k) &= e_R(t_k) - e_R(t_{k-1}) \end{aligned}$$

over the time interval $t \in (t_{k-1}, t_k]$ for $\Delta T_k = t_k - t_{k-1}$.

From the third line of eqn. (9.15), given $\hat{\psi}_{k-1}$, the updated yaw is

$$\begin{aligned} \hat{\psi}_k &= \hat{\psi}_{k-1} + \frac{1}{L}\frac{2\pi}{C}\left(\hat{R}_L \frac{\Delta e_L(k)}{\Delta T_k} - \hat{R}_R \frac{\Delta e_R(k)}{\Delta T_k}\right)\Delta T_k \\ &= \hat{\psi}_{k-1} + \frac{1}{L}\frac{2\pi}{C}\left(\hat{R}_L \Delta e_L(k) - \hat{R}_R \Delta e_R(k)\right). \end{aligned} \tag{9.17}$$

Similarly, from the first two lines of eqn. (9.15), given the last estimate of the position $\hat{\mathbf{p}}_{k-1}$, the updated vehicle position is

$$\begin{aligned} \hat{\mathbf{p}}_k &= \hat{\mathbf{p}}_{k-1} + \begin{bmatrix} \cos(\hat{\psi}_k) \\ \sin(\hat{\psi}_k) \end{bmatrix} \frac{\pi}{C}\left[\hat{R}_l \frac{\Delta e_L(k)}{\Delta T_k} + \hat{R}_r \frac{\Delta e_R(k)}{\Delta T_k}\right]\Delta T_k \\ &= \hat{\mathbf{p}}_{k-1} + \begin{bmatrix} \cos(\hat{\psi}_k) \\ \sin(\hat{\psi}_k) \end{bmatrix} \frac{\pi}{C}\left[\hat{R}_l \Delta e_L(k) + \hat{R}_r \Delta e_R(k)\right]. \end{aligned} \tag{9.18}$$

Due to the integer nature of $\Delta e_L(k)$ and $\Delta e_R(k)$, if the assumptions were correct (i.e., planar surface, no slip, and high update rate), then the results of eqns. (9.17) and (9.18) would be very accurate. Also, eqn. (9.18) could be implemented with either $\hat{\psi}_{k-1}$ or $\frac{1}{2}(\hat{\psi}_{k-1} + \hat{\psi}_k)$ used in place of $\hat{\psi}_k$ or the position update could be performed using both $\hat{\psi}_{k-1}$ and $\hat{\psi}_k$ in a predictor-corrector implementation.

9.4 Error State Dynamic Model

The error state vector is $\delta\mathbf{x} = [\delta n, \delta e, \delta\psi, \delta R_L, \delta R_R]^\top$, where each error term is defined as $\delta x = x - \hat{x}$. In this section, we will also use the following variables to decrease the complexity of the resulting equations:

$$\hat{u}_L = \hat{R}_L\dot{\hat{\phi}}_L, \quad \hat{u}_R = \hat{R}_R\dot{\hat{\phi}}_R, \text{ and } \hat{u} = (\hat{u}_L + \hat{u}_R)/2.$$

Each of these quantities is computable as an average over any time increment based on the encoder readings.

Forming the difference of the Taylor series expansion of eqn. (9.9) and eqn. (9.15) results in the error state equations

$$\begin{aligned}
\delta\dot{n} &= \frac{1}{2}\cos(\hat{\psi})\left(\frac{\hat{u}_L}{\hat{R}_L}\delta R_L + \frac{\hat{u}_R}{\hat{R}_R}\delta R_R\right) - \hat{u}\sin(\hat{\psi})\delta\psi + \omega_n && (9.19)\\
\delta\dot{e} &= \frac{1}{2}\sin(\hat{\psi})\left(\frac{\hat{u}_L}{\hat{R}_L}\delta R_L + \frac{\hat{u}_R}{\hat{R}_R}\delta R_R\right) + \hat{u}\cos(\hat{\psi})\delta\psi + \omega_e && (9.20)\\
\delta\dot{\psi} &= \frac{1}{L}\left(\frac{\hat{u}_L}{\hat{R}_L}\delta R_L - \frac{\hat{u}_R}{\hat{R}_R}\delta R_R\right) + \omega_\psi. && (9.21)
\end{aligned}$$

The error models for R_R and R_L are given by eqns. (9.12–9.13).

The error state equations can be put into be in the standard form of eqn. (4.57) with

$$\mathbf{F}(t) = \begin{bmatrix} 0 & 0 & -\hat{u}\sin(\hat{\psi}) & \frac{\hat{u}_L}{2\hat{R}_L}\cos(\hat{\psi}) & \frac{\hat{u}_R}{2\hat{R}_R}\cos(\hat{\psi}) \\ 0 & 0 & \hat{u}\cos(\hat{\psi}) & \frac{\hat{u}_L}{2\hat{R}_L}\sin(\hat{\psi}) & \frac{\hat{u}_R}{2\hat{R}_R}\sin(\hat{\psi}) \\ 0 & 0 & 0 & \frac{1}{L}\frac{\hat{u}_L}{\hat{R}_L} & -\frac{1}{L}\frac{\hat{u}_R}{\hat{R}_R} \\ 0 & 0 & 0 & -\lambda_R & 0 \\ 0 & 0 & 0 & 0 & -\lambda_R \end{bmatrix} \qquad (9.22)$$

and $\mathbf{\Gamma} = \mathbf{I}_5$ where $\boldsymbol{\omega} = [\omega_n, \omega_e, \omega_\psi, \omega_L, \omega_R]$.

Let σ_1^2, σ_2^2, and σ_3^2 denote the PSD's of ω_n, ω_e, and ω_ψ, respectively. These process driving noise terms are included in the model due to the fact that the assumptions will not be perfectly satisfied. For example, the wheels may slip or the surface may not be perfectly planar. The wheel slip is a function of wheel speed or acceleration. Similarly, the effect of a non-planar surface on the computed position is a function of the speed. Therefore, the quantities σ_1^2, σ_2^2, and σ_3^2 and hence the $\mathbf{Q}$ matrix are sometimes defined to be increasing functions of the speed u and/or the acceleration (i.e. as indicated by the change u).

9.5 GPS Aiding

Using the navigation state vector defined in eqn. (9.14), this section presents the GPS measurement prediction and the GPS measurement residual equations that would be used by the Kalman filter.

Using the estimate of the vehicle position $\hat{\mathbf{p}}$ and the ECEF position of the i-th satellite $\hat{p}^i$, which is computed from the ephemeris data as discussed in Appendix C, the computed range to the i-th satellite vehicle (SV) is

$$\hat{R}(\hat{\mathbf{x}}, \hat{\mathbf{p}}^i) = \left\|\hat{\mathbf{p}}^e - \hat{\mathbf{p}}^i\right\|_2. \qquad (9.23)$$