

Tele-Experiment

Kinematics and Navigation of the MERLIN Robot

Name: Kaung Sett Thu
Matriculation Number: 12503548
Institute: Technische Hochschule Deggendorf
Submission Date: February 14, 2026

Contents

1	The Imaginary Wheel	3
1.1	Imaginary Wheel ϕ in Relation to Outer Wheel ϕ_o	3
1.2	Imaginary Wheel ϕ in Relation to Inner Wheel ϕ_i	4
2	Calculating the Robot Pose using the Odometry (Dead Reckoning)	5
2.1	Pose Estimation in relation to the Velocity v_{CR}	5
2.2	Pose Estimation in relation to Velocity at Imaginary Wheel v_{IM}	8
2.3	Discrete Form of the Robot's Pose in relation to v_{CR}	10
2.4	Recurrent Form of Equations	10
3	Shifting and Rotation	12
3.1	Shifting MERLIN 50 cm to the Right	12
3.2	Rotating MERLIN 90°	13
4	Analysis of the slipping of MERLIN	15

1 The Imaginary Wheel

The relationship between the steering angle of the imaginary wheel ϕ , the wheelbase L , and the radius of the robot R can be described as follow.

$$\tan \phi = \frac{L}{R}$$

1.1 Imaginary Wheel ϕ in Relation to Outer Wheel ϕ_o

The turning radius of the outer wheel from ICR is $R + \frac{D}{2}$. Therefore

$$\tan \phi_o = \frac{L}{R + \frac{D}{2}}$$

$$R + \frac{D}{2} = \frac{L}{\tan \phi_o}$$

$$R = \frac{L}{\tan \phi_o} - \frac{D}{2}$$

Substituting formula for R in terms of steering angle of the outer wheel to formula for the steering angle of the imaginary wheel

$$\tan \phi = \frac{\frac{L}{\tan \phi_o} - \frac{D}{2}}{L}$$

$$\tan \phi = \frac{\frac{1}{\tan \phi_o} - \frac{D}{2L}}{1}$$

$$\tan \phi = \frac{\tan \phi_o}{1 - \frac{D}{2L} \cdot \tan \phi_o}$$

$$\phi = \arctan \left(\frac{\tan \phi_o}{1 - \frac{D}{2L} \cdot \tan \phi_o} \right)$$

1.2 Imaginary Wheel ϕ in Relation to Inner Wheel ϕ_i

The turning radius of the inner wheel from ICR is $R - \frac{D}{2}$. Therefore

$$\tan \phi_i = \frac{L}{R - \frac{D}{2}}$$

$$R - \frac{D}{2} = \frac{L}{\tan \phi_i}$$

$$R = \frac{L}{\tan \phi_i} + \frac{D}{2}$$

Substituting formula for R in terms of steering angle of the inner wheel to formula for the steering angle of the imaginary wheel

$$\tan \phi = \frac{L}{\frac{L}{\tan \phi_i} + \frac{D}{2}}$$

$$\tan \phi = \frac{1}{\frac{1}{\tan \phi_i} + \frac{D}{2L}}$$

$$\tan \phi = \frac{\tan \phi_i}{1 + \frac{D}{2L} \cdot \tan \phi_i}$$

$$\phi = \arctan \left(\frac{\tan \phi_i}{1 + \frac{D}{2L} \cdot \tan \phi_i} \right)$$

2 Calculating the Robot Pose using the Odometry (Dead Reckoning)

2.1 Pose Estimation in relation to the Velocity v_{CR}

The angular velocity of the robot $\omega(t)$ in relation to the velocity of the point CR $v_{CR}(t)$ can be expressed as

$$\omega(t) = \frac{v_{CR}(t)}{R_{CR}}$$

Since v_{CR} is constant, angular velocity ω can be expressed as a constant.

$$\omega = \frac{v_{CR}}{R_{CR}}$$

The relation of the steering angle ϕ to the radius from Instantenous Center of Rotation (ICR) R_{CR} can be rearranged to express it in terms of the radius.

$$\tan \phi = \frac{L}{R_{CR}}$$

$$R_{CR} = \frac{L}{\tan \phi}$$

Replacing the formula for radius R_{CR} into the formula for angular velocity ω ,

$$\begin{aligned} \omega &= \frac{v_{CR}}{\frac{L}{\tan \phi}} \\ &= \frac{v_{CR}}{L} \cdot \tan \phi \end{aligned}$$

The pose estimation for the orientation of the robot θ can be expressed as

$$\begin{aligned} \theta(t) &= \theta_0 + \int_0^t \omega d\tau \\ &= \theta_0 + \omega [\tau]_0^t \\ &= \theta_0 + \omega \cdot t \end{aligned}$$

The pose estimation for x coordinate of the robot can be expressed as

$$\begin{aligned} x(t) &= x_0 + \int_0^t v_{CR} \cdot \cos(\theta(\tau)) d\tau \\ &= x_0 + v_{CR} \int_0^t \cos(\theta_0 + \omega \cdot \tau) d\tau \end{aligned}$$

Integrating $\cos(\theta_0 + \omega \cdot \tau)$ using U-Substitution

Let

$$u = \theta_0 + \omega \cdot \tau$$

$$\frac{du}{d\tau} = \omega$$

$$d\tau = \frac{1}{\omega} \cdot du$$

When $\tau = 0$,

$$u = \theta_0$$

When $\tau = t$,

$$u = \theta_0 + \omega \cdot t$$

$$\begin{aligned} x(t) &= x_0 + v_{CR} \int_{\theta_0}^{\theta_0 + \omega \cdot t} \cos(u) \frac{1}{\omega} \cdot du \\ &= x_0 + \frac{v_{CR}}{\omega} \int_{\theta_0}^{\theta_0 + \omega \cdot t} \cos(u) \cdot du \\ &= x_0 + \frac{v_{CR}}{\omega} \left[\sin(u) \right]_{\theta_0}^{\theta_0 + \omega \cdot t} \\ &= x_0 + \frac{v_{CR}}{\omega} (\sin(\theta_0 + \omega \cdot t) - \sin(\theta_0)) \end{aligned}$$

Replacing the formula for the angular velocity ω into the equation,

$$\begin{aligned} x(t) &= x_0 + v_{CR} \cdot \frac{R_{CR}}{v_{CR}} (\sin(\theta_0 + \omega \cdot t) - \sin(\theta_0)) \\ &= x_0 + R_{CR} (\sin(\theta_0 + \omega \cdot t) - \sin(\theta_0)) \end{aligned}$$

The pose estimation for y coordinate of the robot can be expressed as

$$\begin{aligned} y(t) &= y_0 + \int_0^t v_{CR} \cdot \sin(\theta(\tau)) d\tau \\ &= y_0 + v_{CR} \int_0^t \sin(\theta_0 + \omega \cdot \tau) d\tau \end{aligned}$$

Integrating $\sin(\theta_0 + \omega \cdot \tau)$ using U-Substitution

Let

$$u = \theta_0 + \omega \cdot \tau$$

$$\frac{du}{d\tau} = \omega$$

$$d\tau = \frac{1}{\omega} \cdot du$$

When $\tau = 0$,

$$u = \theta_0$$

When $\tau = t$,

$$u = \theta_0 + \omega \cdot t$$

$$\begin{aligned} y(t) &= y_0 + v_{CR} \int_{\theta_0}^{\theta_0 + \omega \cdot t} \sin(u) \frac{1}{\omega} \cdot du \\ &= y_0 + \frac{v_{CR}}{\omega} \int_{\theta_0}^{\theta_0 + \omega \cdot t} \sin(u) \cdot du \\ &= y_0 + \frac{v_{CR}}{\omega} \left[-\cos(u) \right]_{\theta_0}^{\theta_0 + \omega \cdot t} \\ &= y_0 + \frac{v_{CR}}{\omega} (\cos(\theta_0) - \cos(\theta_0 + \omega \cdot t)) \end{aligned}$$

Replacing the formula for the angular velocity ω into the equation,

$$\begin{aligned} y(t) &= y_0 + v_{CR} \cdot \frac{R_{CR}}{v_{CR}} (\cos(\theta_0) - \cos(\theta_0 + \omega \cdot t)) \\ &= y_0 + R_{CR} (\cos(\theta_0) - \cos(\theta_0 + \omega \cdot t)) \end{aligned}$$

The pose estimation in relation to the velocity v_{CR} can collectively be expressed in a matrix form.

$$p(t) = \begin{bmatrix} x(t) \\ y(t) \\ \theta(t) \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ \theta_0 \end{bmatrix} + \begin{bmatrix} R_{CR}(\sin(\theta_0 + \omega t) - \sin \theta_0) \\ R_{CR}(\cos \theta_0 - \cos(\theta_0 + \omega t)) \\ \omega \cdot t \end{bmatrix}$$

2.2 Pose Estimation in relation to Velocity at Imaginary Wheel v_{IM}

In the right-angled triangle formed by the rear axle center, the imaginary front wheel, and the instantaneous center of rotation (ICR), the distance from the ICR to the imaginary wheel, R_{IM} is the hypotenuse of this triangle, while the wheelbase L corresponds to the side opposite the steering angle ϕ . The steering angle ϕ can therefore be expressed by the use of the *sine* ratio.

$$\sin \phi = \frac{L}{R_{IM}}$$

The equation can be rearranged to express it in terms of the radius R_{IM} .

$$R_{IM} = \frac{L}{\sin \phi}$$

An angular velocity ω can be expressed as follows.

$$\begin{aligned} \omega &= \frac{v_{IM}}{R_{IM}} \\ &= \frac{v_{IM}}{\frac{L}{\sin \phi}} \\ &= \frac{v_{IM} \cdot \sin \phi}{L} \end{aligned}$$

Since the angular velocity ω remains the same for both the rear axle and the front wheel, the relationship between velocity at the rear axis v_{CR} and the imaginary wheel v_{IM} can

be calculated.

$$\frac{v_{CR}}{R_{CR}} = \frac{v_{IM}}{R_{IM}}$$

$$v_{CR} = \frac{v_{IM}}{R_{IM}} \cdot R_{CR}$$

$$v_{CR} = \frac{v_{IM} \cdot \sin \phi}{L} \cdot \frac{L}{\tan \phi}$$

$$v_{CR} = v_{IM} \cdot \cos \phi$$

Using this relationship between v_{CR} and v_{IM} , the pose estimation for $x(t)$ and $y(t)$ can be expressed in terms of v_{IM} as

$$x(t) = x_0 + \int_0^t (v_{IM} \cdot \cos \phi) \cdot \cos(\theta(\tau)) d\tau$$

$$y(t) = y_0 + \int_0^t (v_{IM} \cdot \cos \phi) \cdot \sin(\theta(\tau)) d\tau$$

Since angular velocity ω is constant the pose estimation for orientation $\theta(t)$ remains the same.

Following the same integration steps as section 2.1 for $x(t)$ and $y(t)$, we get

$$\begin{aligned} x(t) &= x_0 + \frac{v_{IM} \cdot \cos \phi}{\omega} (\sin(\theta_0 + \omega \cdot t) - \sin(\theta_0)) \\ &= x_0 + (v_{IM} \cdot \cos \phi) \cdot \frac{L}{v_{IM} \cdot \sin \phi} (\sin(\theta_0 + \omega \cdot t) - \sin(\theta_0)) \\ &= x_0 + \frac{L \cdot \cos \phi}{\sin \phi} (\sin(\theta_0 + \omega \cdot t) - \sin(\theta_0)) \\ &= x_0 + \frac{L}{\tan \phi} (\sin(\theta_0 + \omega \cdot t) - \sin(\theta_0)) \end{aligned}$$

$$\begin{aligned}
y(t) &= y_0 + \frac{v_{IM} \cdot \cos \phi}{\omega} (\cos(\theta_0) - \cos(\theta_0 + \omega \cdot t)) \\
&= y_0 + (v_{IM} \cdot \cos \phi) \cdot \frac{L}{v_{IM} \cdot \sin \phi} (\cos(\theta_0) - \cos(\theta_0 + \omega \cdot t)) \\
&= y_0 + \frac{L \cdot \cos \phi}{\sin \phi} (\cos(\theta_0) - \cos(\theta_0 + \omega \cdot t)) \\
&= y_0 + \frac{L}{\tan \phi} (\cos(\theta_0) - \cos(\theta_0 + \omega \cdot t))
\end{aligned}$$

Since $\frac{L}{\tan \phi}$ is radius from rear axle R_{CR} , we can see that pose estimation in relation to velocity at imaginary wheel v_{IM} has a similar matrix to pose estimation in relation to velocity at rear axle v_{CR}

$$p(t) = \begin{bmatrix} x(t) \\ y(t) \\ \theta(t) \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ \theta_0 \end{bmatrix} + \begin{bmatrix} R_{CR}(\sin(\theta_0 + \omega t) - \sin \theta_0) \\ R_{CR}(\cos \theta_0 - \cos(\theta_0 + \omega t)) \\ \omega \cdot t \end{bmatrix}$$

with angular velocity ω being expressed in terms of velocity at imaginary wheel v_{IM} as

$$\omega = \frac{v_{IM} \cdot \sin \phi}{L}$$

2.3 Discrete Form of the Robot's Pose in relation to v_{CR}

Based on the matrix derived in section 2.1, assuming constant velocity v_{CR} and constant steering angle ϕ over the sampling interval Δt , the exact discrete-time update of the robot pose can be expressed as

$$p(k+1) = \begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} + \begin{bmatrix} R_{CR}(\sin(\theta_k + \omega \Delta t) - \sin \theta_k) \\ R_{CR}(\cos \theta_k - \cos(\theta_k + \omega \Delta t)) \\ \omega \cdot \Delta t \end{bmatrix}$$

2.4 Recurrent Form of Equations

For recurrent odometry updates under constant steering angle and velocity over the time Δt , the pose can be updated recurrently by following pipeline of equations

1. Update the orientation

$$\theta_{k+1} = \theta_k + \frac{v_{CR} \cdot \tan \phi}{L} \cdot \Delta t$$

2. Update the x coordinate

$$x_{k+1} = x_k + \frac{L}{\tan \phi} (\sin \theta_{k+1} - \sin \theta_k)$$

3. Update the y coordinate

$$y_{k+1} = y_k + \frac{L}{\tan \phi} (\cos \theta_k - \cos \theta_{k+1})$$

In case of a straight line motion where ϕ is 0, the equations simplify to

1. Update the orientation

$$\theta_{k+1} = \theta_k$$

2. Update the x coordinate

$$x_{k+1} = x_k + v_{CR} \cdot \cos \theta_k \cdot \Delta t$$

3. Update the y coordinate

$$y_{k+1} = y_k + v_{CR} \cdot \sin \theta_k \cdot \Delta t$$

3 Shifting and Rotation

3.1 Shifting MERLIN 50 cm to the Right

Wheelbase (L): 230 mm

Max steering angle (ϕ_{\max}): 14°

Maximum steering angle is chosen to save as much longitudinal distance as possible.

Turning radius (R):

$$\begin{aligned} R &= \frac{L}{\tan \phi_{\max}} \\ &= \frac{230 \text{ mm}}{\tan 14^\circ} \\ &= 92.25 \text{ cm} \end{aligned}$$

Lateral displacement ($dist$): 50 cm

Since $dist < 2R$, Maneuver A will be used.

$$\begin{aligned} \theta &= \arccos \left(\frac{2R - dist}{2R} \right) \\ &= \arccos \left(\frac{2 \cdot 92.25 \text{ cm} - 50 \text{ cm}}{2 \cdot 92.25 \text{ cm}} \right) \\ &= 0.754 \text{ rad} = 43.2^\circ \end{aligned}$$

$$\begin{aligned} d_1 &= d_3 = R \cdot \theta \\ &= 92.25 \text{ cm} \cdot 0.754 \text{ rad} \\ &= 69.56 \text{ cm} \end{aligned}$$

$$d_2 = 0$$

$$\begin{aligned}
d_4 &= 2 \cdot \sqrt{R^2 - \left(R - \frac{dist}{2}\right)^2} \\
&= 2 \cdot \sqrt{(92.25 \text{ cm})^2 - \left(92.25 \text{ cm} - \frac{50 \text{ cm}}{2}\right)^2} \\
&= 126.3 \text{ cm}
\end{aligned}$$

The set of maneuvers the robot should take is as follows

1. $\phi = -14^\circ$ (towards right) for 69.56 cm
2. $\phi = 14^\circ$ (towards left) for 69.56 cm

No straight segment needed. The robot needs at least 146.3 cm (126.3 cm + 20 cm for clearance) in front of it to clearly shift 50 cm right.

3.2 Rotating MERLIN 90°

Wheelbase (L): 230 mm

Max steering angle (ϕ_{\max}): 14°

Maximum steering angle is chosen to save as much longitudinal distance as possible.

Turning radius (R):

$$\begin{aligned}
R &= \frac{L}{\tan \phi_{\max}} \\
&= \frac{230 \text{ mm}}{\tan 14^\circ} \\
&= 92.25 \text{ cm}
\end{aligned}$$

Angle to rotate (α): $90^\circ = \frac{\pi}{2} \text{ rad}$

$$\begin{aligned}
\theta &= \arccos\left(\frac{\alpha}{2}\right) \\
&= \arccos\left(\frac{\pi/2}{2}\right) \\
&= 0.6675 \text{ rad} = 38.24^\circ
\end{aligned}$$

$$\begin{aligned}
d_1 &= d_3 = R \cdot \theta \\
&= 92.25 \text{ cm} \cdot 0.6675 \text{ rad} \\
&= 61.58 \text{ cm} \\
\\
d_2 &= 2R \cdot \tan \theta \\
&= 2 \cdot (92.25 \text{ cm}) \cdot \tan(0.6675 \text{ rad}) \\
&= 145.39 \text{ cm}
\end{aligned}$$

The set of maneuvers the robot should take is as follows

1. $\phi = -14^\circ$ (towards right) for 61.58 cm
2. $\phi = 14^\circ$ (towards left) for 61.58 cm

No straight segment needed. The robot needs at least 165.39 cm (145.39 cm + 20 cm for clearance) in front of it to clearly rotate 90°.

4 Analysis of the slipping of MERLIN

To analyse the slipping of MERLIN on the straight line motion,

1. Initial set-up
 - The robot is aligned with the X axis i.e $\theta_{init} = 0$
 - Robot parameters
 - Wheel diameter: $d_{wheel} = 100\text{ mm}$
 - Encoder resolution: $c = 1024$
 - Wheelbase: $D = 230\text{ mm}$
2. Experiment procedure
 - Drive 10 times in a straight line for 1 m
3. Data collection methods
 - Encoder values N_R and N_L to calculate distance and heading
 - IMU to integrate heading θ_{gyro}
 - Tracking server to record actual x , y and z using IR Sensors
4. Identify systematic and non-systematic errors
 - The mean values of the left and right encoder readings help identify systematic errors
 - High deviations in the runs identify random slippage and other non-systematic errors
5. Calculation of control odometry
 - The control coordinates of the MERLIN robot can be calculated from the

encoder values using the formulas

$$x = \frac{\pi \cdot d_{wheel}}{2c} (N_R + N_L) \cdot \cos \left(\frac{\pi \cdot d_{wheel}}{cD} (N_R - N_L) \right)$$

$$y = \frac{\pi \cdot d_{wheel}}{2c} (N_R + N_L) \cdot \sin \left(\frac{\pi \cdot d_{wheel}}{cD} (N_R - N_L) \right)$$

$$\theta = \frac{\pi \cdot d_{wheel}}{cD} (N_R - N_L)$$

- These values are the control values

6. Measurement of actual values

- IMU to measure the final heading of the robot
- Tracking server to get the actual x and y coordinate of the robot

7. Compare control values and actual values

- Comparing calculated x , y , and θ and the measured x , y , and θ gives the slipping of the MERLIN robot
- Use (x_{ctrl}, y_{ctrl}) and (x_{act}, y_{act}) coordinates to calculate the longitudinal slip-page
- The difference between θ_{act} and θ_{ctrl} gives the orientation error