COMP0130 Robot Vision and Navigation
Coursework 1: Integrated Navigation for a Robotic Lawnmower

Yuang DU, 23068251, ucab190@ucl.ac.uk
Yufeng WU, 23083138, yufeng.wu.22@ucl.ac.uk
Yujie WANG, 23087003, ucab211@ucl.ac.uk

February 8, 2024

Method

The task is to use the sensors data to compute the best possible horizontal position, horizontal velocity, and heading solution for the lawnmower at each point in time. To accomplish this task, our group implemented a total of five steps:

1. Determine the initial position and velocity of the lawnmower based on iterative least squares using data from GNSS.
2. Determine the position and velocity of the lawnmower at each time based on Kalman filtering using data from GNSS.
3. Generate the corrected heading of the lawnmower based on Kalman filtering using data from a gyroscope and magnetic compass.
4. Model the speed model. Determine the position and the velocity of the lawnmower at each time using data from wheel speed sensors with the Dead-Reckoning approach.
5. Fuse the data from step 2 and 4 based on Kalman filtering and integrate DR/GNSS to get the final result.

The overall flow chart is shown in Fig 1. Detailed descriptions for each step are given below.

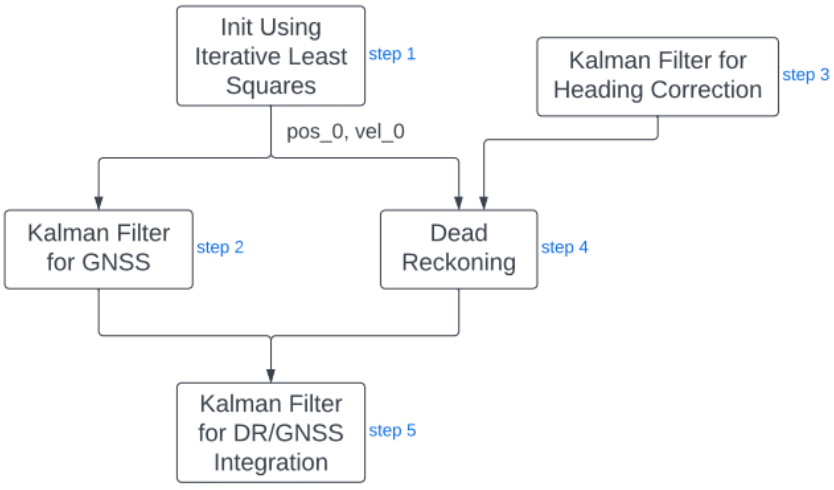

Figure 1: Flow Chart of the Work

1. Initializing Position and Velocity

Before applying the Kalman filter, we need to determine the initial position and velocity of the lawnmower. This is solved by iterative least squares, referring to **Task 1a**, **Task 1b**, and **Task 4** in Workshop 1[1].

Assume the lawnmower's initial position is at the centre of the Earth (Cartesian ECEF coordinates: 0, 0, 0) and the initial velocity is (0, 0, 0). Then iterate the least-squares position and velocity computation, using the solution from the previous iteration as the initial position and velocity for subsequent iterations. Within each iteration step, calculations are implemented based on the following equations:

1. Update the position ($\hat{\mathbf{r}}_{ea}^{e-}$) and velocity ($\hat{\mathbf{v}}_{ea}^{e-}$) of the lawnmower at the beginning of current iteration from the results of the previous iteration, and calculate the position ($\hat{\mathbf{r}}_{ej}^{e}$) and velocity ($\hat{\mathbf{v}}_{ej}^{e}$) of each satellite at $t = 0$.

2. Predict the ranges from the lawnmower (a) to each satellite (j) using

$$\begin{aligned} \hat{r}_{aj}^{-} &= \sqrt{[\mathbf{C}_e^I \hat{\mathbf{r}}_{ej}^{e} - \hat{\mathbf{r}}_{ea}^{e-}]^T [\mathbf{C}_e^I \hat{\mathbf{r}}_{ej}^{e} - \hat{\mathbf{r}}_{ea}^{e-}]} \\ \mathbf{C}_e^I &\approx \begin{bmatrix} 1 & \omega_{ie} r_{aj}/c & 0 \\ -\omega_{ie} r_{aj}/c & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{1}$$

   where the Earth rotation rate (ω_{ie}) is 7.292115×10^{-5}rad s^{-1} and the speed of light (c) is 299792458m s^{-1}.

3. Compute the line-of-sight unit vector from the lawnmower to each satellite using

$$\mathbf{u}_{aj}^{e} = \frac{\mathbf{C}_e^I \hat{\mathbf{r}}_{ej}^{e} - \hat{\mathbf{r}}_{ea}^{e-}}{\hat{r}_{aj}^{-}} \tag{2}$$

4. Predict the range rates from the lawnmower (a) to each satellite (j) using

$$\dot{\hat{r}}_{aj}^{-} = \hat{\mathbf{u}}_{aj}^{e-^T} [\mathbf{C}_e^I (\hat{\mathbf{v}}_{ej}^{e} + \mathbf{\Omega}_{ie}^{e} \hat{\mathbf{r}}_{ej}^{e}) - (\hat{\mathbf{v}}_{ea}^{e-} + \mathbf{\Omega}_{ie}^{e} \hat{\mathbf{r}}_{ea}^{e-})] \tag{3}$$

   where $\mathbf{\Omega}_{ie}^{e}$ is the skew symmetric matrix of the Earth rotation rate.

5. Formulate the predicted position state vector ($\hat{\mathbf{x}}^{-}$), velocity state vector ($\dot{\hat{\mathbf{x}}}^{-}$), measurement position innovation vector ($\delta\mathbf{z}^{-}$), measurement velocity innovation vector ($\delta\dot{\mathbf{z}}^{-}$) and measurement matrix ($\mathbf{H}_G^e$):

$$\hat{\mathbf{x}}^{-} = \begin{bmatrix} \hat{\mathbf{r}}_{ea}^{e-} \\ \delta\hat{\rho}_c^{a-} \end{bmatrix}, \dot{\hat{\mathbf{x}}}^{-} = \begin{bmatrix} \hat{\mathbf{v}}_{ea}^{e-} \\ \delta\hat{\dot{\rho}}_c^{a-} \end{bmatrix}$$

$$\delta\mathbf{z}^{-} = \begin{bmatrix} \tilde{\rho}_a^5 - \hat{r}_{a5}^{-} - \delta\hat{\rho}_c^{a-} \\ \tilde{\rho}_a^6 - \hat{r}_{a6}^{-} - \delta\hat{\rho}_c^{a-} \\ \tilde{\rho}_a^7 - \hat{r}_{a7}^{-} - \delta\hat{\rho}_c^{a-} \\ \tilde{\rho}_a^9 - \hat{r}_{a9}^{-} - \delta\hat{\rho}_c^{a-} \\ \tilde{\rho}_a^{10} - \hat{r}_{a10}^{-} - \delta\hat{\rho}_c^{a-} \\ \tilde{\rho}_a^{11} - \hat{r}_{a11}^{-} - \delta\hat{\rho}_c^{a-} \\ \tilde{\rho}_a^{15} - \hat{r}_{a15}^{-} - \delta\hat{\rho}_c^{a-} \\ \tilde{\rho}_a^{30} - \hat{r}_{a30}^{-} - \delta\hat{\rho}_c^{a-} \end{bmatrix}, \delta\dot{\mathbf{z}}^{-} = \begin{bmatrix} \tilde{\dot{\rho}}_a^5 - \dot{\hat{r}}_{a5}^{-} - \delta\hat{\dot{\rho}}_c^{a-} \\ \tilde{\dot{\rho}}_a^6 - \dot{\hat{r}}_{a6}^{-} - \delta\hat{\dot{\rho}}_c^{a-} \\ \tilde{\dot{\rho}}_a^7 - \dot{\hat{r}}_{a7}^{-} - \delta\hat{\dot{\rho}}_c^{a-} \\ \tilde{\dot{\rho}}_a^9 - \dot{\hat{r}}_{a9}^{-} - \delta\hat{\dot{\rho}}_c^{a-} \\ \tilde{\dot{\rho}}_a^{10} - \dot{\hat{r}}_{a10}^{-} - \delta\hat{\dot{\rho}}_c^{a-} \\ \tilde{\dot{\rho}}_a^{11} - \dot{\hat{r}}_{a11}^{-} - \delta\hat{\dot{\rho}}_c^{a-} \\ \tilde{\dot{\rho}}_a^{15} - \dot{\hat{r}}_{a15}^{-} - \delta\hat{\dot{\rho}}_c^{a-} \\ \tilde{\dot{\rho}}_a^{30} - \dot{\hat{r}}_{a30}^{-} - \delta\hat{\dot{\rho}}_c^{a-} \end{bmatrix}, \mathbf{H}_G^e = \begin{bmatrix} -u_{a5,x}^e & -u_{a5,y}^e & -u_{a5,z}^e & 1 \\ -u_{a6,x}^e & -u_{a6,y}^e & -u_{a6,z}^e & 1 \\ -u_{a7,x}^e & -u_{a7,y}^e & -u_{a7,z}^e & 1 \\ -u_{a9,x}^e & -u_{a9,y}^e & -u_{a9,z}^e & 1 \\ -u_{a10,x}^e & -u_{a10,y}^e & -u_{a10,z}^e & 1 \\ -u_{a11,x}^e & -u_{a11,y}^e & -u_{a11,z}^e & 1 \\ -u_{a15,x}^e & -u_{a15,y}^e & -u_{a15,z}^e & 1 \\ -u_{a30,x}^e & -u_{a30,y}^e & -u_{a30,z}^e & 1 \end{bmatrix} \tag{4}$$

   where $\tilde{\rho}_a^j$ and $\tilde{\dot{\rho}}_a^j$ is the measured pseudo range and range rate from satellite j to the lawnmower, respectively. $\delta\hat{\rho}_c^{a-}$ and $\delta\hat{\dot{\rho}}_c^{a-}$ is the predicted receiver clock offset for position and velocity, respectively.

6. Compute the position and velocity as well as the clock offsets using unweighted least-squares:

$$\begin{aligned}\begin{bmatrix}\hat{\mathbf{r}}_{ea}^{e+}\\ \delta\hat{\rho}_c^{a+}\end{bmatrix} &= \hat{\mathbf{x}}^+ = \hat{\mathbf{x}}^- + (\mathbf{H}_G^{e^T}\mathbf{H}_G^e)^{-1}\mathbf{H}_G^{e^T}\delta\mathbf{z}^-\\ \begin{bmatrix}\hat{\mathbf{v}}_{ea}^{e+}\\ \delta\hat{\dot{\rho}}_c^{a+}\end{bmatrix} &= \hat{\dot{\mathbf{x}}}^+ = \hat{\dot{\mathbf{x}}}^- + (\mathbf{H}_G^{e^T}\mathbf{H}_G^e)^{-1}\mathbf{H}_G^{e^T}\delta\dot{\mathbf{z}}^-\end{aligned} \tag{5}$$

7. Compute the difference between the result at the current iteration and the result at the previous iteration. Terminate the iteration when the difference is less than 0.001, and let that position and velocity be the initial data about the lawnmower for Kalman filtering.

2. Kalman Filtering for GNSS

Apply Kalman Filter on `Pseudo_ranges.csv` and `Pseudo_range_rates.csv` to determine the position and velocity of the lawnmower at each timestep t. Also, apply outlier detection to remove the data from the satellite which has large noise. This is implemented with reference to **Task 3** in Workshop 1[1] and **Task 2A**, **Task 2B** in Workshop 2[2].

From the GNSS CSV file, we can observe that the satellite monitored a total of 851 data: from $t = 0$ to $t = 425$ with the propagation interval $\tau_s = 0.5$ s. For every timestep t, we compute 8 states as a vector $x = [\mathbf{r}_{ea}^e \quad \mathbf{v}_{ea}^e \quad \delta\rho_c^a \quad \delta\dot{\rho}_c^a]^T$ to get the position, velocity, clock offset and clock drift of the lawnmower. For $t = 0$, we have got the result from step 1 and set it as the initial state input of the Kalman Filter. For the initial error covariance matrix, assuming the initial state uncertainty are 10 m for position and clock offset and 0.1 m/s for velocity and clock drift, the error covariance matrix is initialised as below:

$$\mathbf{P}_0 = \begin{bmatrix}100 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 100 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 100 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0.01 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0.01 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0.01 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 100 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.01\end{bmatrix} \tag{6}$$

After the initialisation, we loop with time t and apply Kalman Filter to get the results of the rest timesteps based on the following equations:

1. Compute the transition matrix and system noise covariance matrix:

$$\mathbf{\Phi}_{k-1} = \begin{bmatrix}\mathbf{I}_3 & \tau_s\mathbf{I}_3 & \mathbf{0}_{3,1} & \mathbf{0}_{3,1}\\ \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_{3,1} & \mathbf{0}_{3,1}\\ \mathbf{0}_{1,3} & \mathbf{0}_{1,3} & 1 & \tau_s\\ \mathbf{0}_{1,3} & \mathbf{0}_{1,3} & 0 & 1\end{bmatrix}, \mathbf{Q}_{k-1} = \begin{bmatrix}\frac{1}{3}S_a\tau_s^3\mathbf{I}_3 & \frac{1}{2}S_a\tau_s^2\mathbf{I}_3 & \mathbf{0}_{3,1} & \mathbf{0}_{3,1}\\ \frac{1}{2}S_a\tau_s^2\mathbf{I}_3 & S_a\tau_s\mathbf{I}_3 & \mathbf{0}_{3,1} & \mathbf{0}_{3,1}\\ \mathbf{0}_{1,3} & \mathbf{0}_{1,3} & S_{c\phi}^a\tau_s + \frac{1}{3}S_{cf}^a\tau_s^3 & \frac{1}{2}S_{cf}^a\tau_s^2\\ \mathbf{0}_{1,3} & \mathbf{0}_{1,3} & \frac{1}{2}S_{cf}^a\tau_s^2 & S_{cf}^a\tau_s\end{bmatrix} \tag{7}$$

where $\tau_s = 0.5$ s, acceleration PSD $S_a^e = 0.01\text{m}^2\text{s}^{-3}$, clock phase PSD $S_{c\phi}^a = 0.01\text{m}^2\text{s}^{-3}$ and clock frequency PSD $S_{cf}^a = 0.04\text{m}^2\text{s}^{-3}$.

2. Propagate the state estimates and error covariance matrix:

$$\begin{aligned}\hat{\mathbf{x}}_k^- &= \mathbf{\Phi}_{k-1}\hat{\mathbf{x}}_{k-1}^+\\ \mathbf{P}_k^- &= \mathbf{\Phi}_{k-1}\mathbf{P}_{k-1}^+\mathbf{\Phi}_{k-1}^T + \mathbf{Q}_{k-1}\end{aligned} \tag{8}$$

3. Predict the ranges $\hat{r}_{aj}^{-}$, unit vector $\mathbf{u}_{aj}^{e}$ and range rates $\hat{\dot{r}}_{aj}^{-}$ using Equ 1, Equ 2 and Equ 3 respectively.

4. Formulate the measurement matrix $\mathbf{H}_k$, measurement noise covariance matrix $\mathbf{R}_k$ and measurement innovation vector $\delta\mathbf{z}_k^{-}$:

$$\mathbf{H}_k = \begin{bmatrix} -u_{a5,x}^{e} & -u_{a5,y}^{e} & -u_{a5,z}^{e} & 0 & 0 & 0 & 1 & 0 \\ -u_{a6,x}^{e} & -u_{a6,y}^{e} & -u_{a6,z}^{e} & 0 & 0 & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -u_{a30,x}^{e} & -u_{a30,y}^{e} & -u_{a30,z}^{e} & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -u_{a5,x}^{e} & -u_{a5,y}^{e} & -u_{a5,z}^{e} & 0 & 1 \\ 0 & 0 & 0 & -u_{a6,x}^{e} & -u_{a6,y}^{e} & -u_{a6,z}^{e} & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & -u_{a30,x}^{e} & -u_{a30,y}^{e} & -u_{a30,z}^{e} & 0 & 1 \end{bmatrix}_{16\times 8}$$

$$\mathbf{R}_k = \begin{bmatrix} \sigma_\rho^2 \times \mathbf{I}_{8,8} & \mathbf{0}_{8,8} \\ \mathbf{0}_{8,8} & \sigma_r^2 \times \mathbf{I}_{8,8} \end{bmatrix}, \delta\mathbf{z}_k^{-} = \begin{bmatrix} \tilde{\rho}_a^5 - \hat{r}_{a5}^{-} - \delta\hat{\rho}_c^{a-} \\ \tilde{\rho}_a^6 - \hat{r}_{a6}^{-} - \delta\hat{\rho}_c^{a-} \\ \vdots \\ \tilde{\rho}_a^{30} - \hat{r}_{a30}^{-} - \delta\hat{\rho}_c^{a-} \\ \tilde{\dot{\rho}}_a^5 - \hat{\dot{r}}_{a5}^{-} - \delta\hat{\dot{\rho}}_c^{a-} \\ \tilde{\dot{\rho}}_a^6 - \hat{\dot{r}}_{a6}^{-} - \delta\hat{\dot{\rho}}_c^{a-} \\ \vdots \\ \tilde{\dot{\rho}}_a^{30} - \hat{\dot{r}}_{a30}^{-} - \delta\hat{\dot{\rho}}_c^{a-} \end{bmatrix} \tag{9}$$

   where $\sigma_\rho = 10$m and $\sigma_r = 0.05$m/s.

5. Compute the Kalman gain matrix:

$$\mathbf{K}_k = \mathbf{P}_k^{-}\mathbf{H}_k^{T}(\mathbf{H}_k\mathbf{P}_k^{-}\mathbf{H}_k^{T} + \mathbf{R}_k)^{-1} \tag{10}$$

6. Update the state estimates and error covariance matrix:

$$\begin{aligned} \hat{\mathbf{x}}_k^{+} &= \hat{\mathbf{x}}_k^{-} + \mathbf{K}_k\delta\mathbf{z}_k^{-} \\ \mathbf{P}_k^{+} &= (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^{-} \end{aligned} \tag{11}$$

   This 2 values are also the input values of $\hat{\mathbf{x}}_{k-1}^{-}$ and $\mathbf{P}_{k-1}^{-}$ for the next loop.

At the same time, Outlier Detection is also integrated from step 3 to 5 of the Kalman Filter. Compute the residuals vector and residuals covariance matrix:

$$\begin{aligned} \mathbf{v} &= [\mathbf{H}_k(\mathbf{H}_k^{T}\mathbf{H}_k)^{-1}\mathbf{H}_k^{T} - \mathbf{I}]\delta\mathbf{z}_k^{-} \\ \mathbf{C}_v &= [\mathbf{I} - \mathbf{H}_k(\mathbf{H}_k^{T}\mathbf{H}_k)^{-1}\mathbf{H}_k^{T}]\sigma^2 \end{aligned} \tag{12}$$

where the measurement error standard deviation $\sigma = 5$m. And the satellite j is an outlier when the following condition is met:

$$|v_j| > \sqrt{C_{vjj}}T \tag{13}$$

where C_{vjj} is the j^{th} diagonal element of $\mathbf{C}_v$ and the outlier detection threshold $T = 6$. If an outlier is detected, recalculate related variables from step 3 to 5 of the Kalman Filter. Until no new outlier is detected at that epoch, forward to the rest step. It is also noted that outlier detection is applied at every loop epoch.

3. Kalman Filtering for Heading Correction

This part performs data fusion of a gyroscope and a magnetic compass with a Kalman filter to correct the heading solution of the gyroscope. The heading solution can be obtained from the readings of the magnetic compass directly or the integral of the angular rate measurements from the gyroscope, which are recorded as column 7 and column 6 in the file `Dead_reckoning.csv`. Integrating the data from these two sensors combines the long-term accuracy of the magnetic heading and the short-term stability of the gyroscope, and the adaptation of the Kalman filter method taking more features of the sensors compared to the smoothing algorithm based on weight allocation, leading to a more physically accurate solution.

The architecture of the integration is as described in the lecture 3B slides, page 39 [3]. The pipeline is the same as step 2, but some state vectors and matrices are different.

1. Define the states and state estimation error covariance matrix:

   The state of the Kalman filter is a combination of the gyro-derived heading error $\delta\psi^g$ and the gyroscope bias $\mathbf{b}_g$:

   $$\mathbf{x} = \begin{bmatrix} \delta\psi^g \\ \mathbf{b}_g \end{bmatrix} \tag{14}$$

   The initial states are set to zeros since the error and the bias are unknown. The state estimation error covariance matrix $\mathbf{P}_0$ is initialised as:

   $$\mathbf{P}_0 = \begin{bmatrix} \sigma_{gn}^2 & 0 \\ 0 & \sigma_{gb}^2 \end{bmatrix} \tag{15}$$

   where σ_{gn} and the σ_{gn} are the random noise std deviation and the bias std deviation of the gyroscope provided [4].

2. Compute the state transition matrix and the system noise covariance matrix:

   Since the heading error is the integral of the gyro bias, the state transition matrix $\mathbf{\Phi}_{k-1}$ is:

   $$\mathbf{\Phi}_{k-1} = \begin{bmatrix} 1 & \tau_s \\ 0 & 1 \end{bmatrix} \tag{16}$$

   The system noise covariance matrix is composed of both the randome noise and the bias, therefore:

   $$\mathbf{Q}_{k-1} = \begin{bmatrix} \mathbf{S}_{rg}\tau_s + \frac{1}{3}\mathbf{S}_{bgd}\tau_s^3 & \frac{1}{2}\mathbf{S}_{bgd}\tau_s^2 \\ \frac{1}{2}\mathbf{S}_{bgd}\tau_s^2 & \mathbf{S}_{bgd}\tau_s \end{bmatrix} \tag{17}$$

   where $\mathbf{S}_{rg}$ is the gyro random noise PSD provided [4] and $\mathbf{S}_{bgd}$ is the gyro bias PSD estimated as $2 \times 10^{-7}\,\text{rad}^2$.

3. Propagate the state estimates and error covariance matrix with Equ 8.

4. Formulate the measurement matrix $\mathbf{H}_k$, measurement noise covariance matrix $\mathbf{R}_k$ and measurement innovation vector δz_k^-:

   The Kalman filter measurement is the subtraction of the heading solution from the compass Ψ_k^M and the gyroscope Ψ_k^g, which is denoted as:

   $$\tilde{Z}_k = \Psi_k^M - \Psi_k^g \tag{18}$$

   where Ψ_k^g is then derived by integrating gyroscope angular rate measurement as described in the lecture 3A slides, page 53 [5]:

   $$\Psi_k^g = \Psi_{k-1}^g + \omega_k\tau_s \tag{19}$$

   where ω_k is the measurement from the gyroscope. The heading from the magnetic compass is used as the initial heading.

The measurement matrix $\mathbf{H}_k$ is $[-\mathbf{1} \quad \mathbf{0}]$ and the measurement innovation matrix δz_k^- is:

$$\delta z_k^- = \tilde{Z}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^- \tag{20}$$

The measurement noise covariance $\mathbf{R}_k$ is the magnetic heading noise variance $\sigma_{\mathbf{M}}^{\mathbf{2}}$, which is provided by the coursework instruction [4].

5. Compute the Kalman gain matrix with Equ 10.
6. Update the state estimates and error covariance matrix with Equ 11.
7. At the end of each epoch, the corrected heading Ψ_k^C is obtained by subtracting the error from the original gyro-derived heading:

$$\Psi_k^C = \Psi_k^g - \delta\psi^g \tag{21}$$

4. Lawnmower Dead Reckoning

This section obtains the position and velocity of the lawnmower in the NED frame with the Dead-reckoning method with reference to **Task 1** in Workshop 3[6]. In contrast to Workshop 3, the data provided related to velocity is the speed measurement of each wheel. Therefore, a kinematics model described in Kozlowski's paper [7] is applied to calculate the average speed at the antenna with the formulas below:

$$v_a = \sqrt{v_f^2 + v_l^2} \tag{22}$$

where v_f is the forward component of the velocity, derived by averaging the speed of the driving wheels, which are the rear wheels declared in the coursework instruction [4]. The v_l is the lateral velocity caused by the vehicle yawing around its instantaneous centre of rotation (ICR), which can be calculated as follows:

$$v_l = \frac{s_{rl} + s_{rr}}{d_{lf}} \times d_{ICR}^a \tag{23}$$

where s_{rl} and s_{rr} are the speed of the rear left wheel and the rear right wheel, d_{lf} is the distance between the left and right wheels (provided as 0.5 m), and d_{ICR}^a is the distance between ICR and the location of the antenna. Since the ICR is unknown, it is assumed that the car is rotating around its centre of geometry, therefore d_{ICR}^a can be calculated with provided data as 0.05 m. The initial position and velocity in the NED frame adapts the solution generated with iterative least-squares in step 1. From $t = 2$, the Dead-reckoning pipeline operates with the following steps:

1. Calculate the north and east velocity with the following equation:

$$\begin{pmatrix} \bar{\mathbf{v}}_{N,k} \\ \bar{\mathbf{v}}_{E,k} \end{pmatrix} = \frac{1}{2} \begin{pmatrix} \cos\psi_k + \cos\psi_{k-1} \\ \sin\psi_k + \sin\psi_{k-1} \end{pmatrix} \bar{\mathbf{v}}_k \tag{24}$$

   where ψ_k is the integrated heading solution from step 3, and $\bar{\mathbf{v}}_k$ is the average speed at the antenna derived by the velocity model above.

2. Calculate the latitude L_k and the longitude λ_k with the following equations [6]:

$$\begin{aligned} L_k &= L_{k-1} + \frac{\bar{\mathbf{v}}_{N,k}(t_k - t_{k-1})}{R_N + h} \\ \lambda_k &= \lambda_{k-1} + \frac{\bar{\mathbf{v}}_{E,k}(t_k - t_{k-1})}{(R_E + h)\cos L_k} \end{aligned} \tag{25}$$

   where h is the geodetic height at each epoch, using the value obtained by the GNSS Kalman filter in step 2. R_N is the meridian radius of curvature and R_E is the transverse radius of curvature obtained by the provided function.

3. Compute the damped instantaneous DR velocity at each epoch:

$$\begin{aligned} \mathbf{v}_{N,k} &= 1.7\bar{\mathbf{v}}_{N,k} - 0.7\bar{\mathbf{v}}_{N,k-1} \\ \mathbf{v}_{E,k} &= 1.7\bar{\mathbf{v}}_{E,k} - 0.7\bar{\mathbf{v}}_{E,k-1} \end{aligned} \tag{26}$$

5. Kalman Filtering for DR/GNSS Integration

This part is to fuse GNSS and DR together to get more precise results. This is solved by Kalman Filter with reference to **Task 2** in Workshop 3[6].

The pipeline is the same as step 2, but some state vectors and matrices are different.

1. The state vector and estimation error covariance matrix:

$$\mathbf{x} = \begin{bmatrix} \delta v_n & \delta v_e & \delta L & \delta \lambda \end{bmatrix}^T$$
$$\mathbf{P}_0^+ = \begin{bmatrix} \sigma_v^2 & 0 & 0 & 0 \\ 0 & \sigma_v^2 & 0 & 0 \\ 0 & 0 & \frac{\sigma_r^2}{(R_N+h_0)^2} & 0 \\ 0 & 0 & 0 & \frac{\sigma_r^2}{(R_E+h_0)^2 \cos^2 L_0} \end{bmatrix} \tag{27}$$

   where the 4 states are north and east DR velocity error, DR latitude error, and DR longitude error, respectively. The initial velocity uncertainty is $\sigma_v = 0.1$ m/s and the initial position uncertainty is $\sigma_r = 10$ m.

2. The transition matrix and system noise covariance matrix:

$$\mathbf{\Phi}_{k-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \frac{\tau_s}{R_N+h_{k-1}} & 0 & 1 & 0 \\ 0 & \frac{\tau_s}{(R_E+h_{k-1})\cos L_{k-1}} & 0 & 1 \end{bmatrix}$$
$$\mathbf{Q}_{k-1} = \begin{bmatrix} S_{DR}\tau_s & 0 & \frac{1}{2}\frac{S_{DR}\tau_s^2}{R_N+h_{k-1}} & 0 \\ 0 & S_{DR}\tau_s & 0 & \frac{1}{2}\frac{S_{DR}\tau_s^2}{(R_E+h_{k-1})\cos L_{k-1}} \\ \frac{1}{2}\frac{S_{DR}\tau_s^2}{R_N+h_{k-1}} & 0 & \frac{1}{3}\frac{S_{DR}\tau_s^3}{(R_N+H_{k-1})^2} & 0 \\ 0 & \frac{1}{2}\frac{S_{DR}\tau_s^2}{(R_E+h_{k-1})\cos L_{k-1}} & 0 & \frac{1}{3}\frac{S_{DR}\tau_s^3}{(R_E+h_{k-1})^2 \cos^2 L_{k-1}} \end{bmatrix} \tag{28}$$

   where the propagation interval is $\tau_s = 0.5$ s and the DR velocity error PSD is $S_{DR} = 0.2\text{m}^2\text{s}^{-3}$.

3. The measurement matrix and measurement noise covariance matrix:

$$\mathbf{H}_k = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix}, \mathbf{R}_k = \begin{bmatrix} \frac{\sigma_{Gr}^2}{(R_N+h_k)^2} & 0 & 0 & 0 \\ 0 & \frac{\sigma_{Gr}^2}{(R_E+h_k)^2 \cos^2 L_k} & 0 & 0 \\ 0 & 0 & \sigma_{Gv}^2 & 0 \\ 0 & 0 & 0 & \sigma_{Gv}^2 \end{bmatrix} \tag{29}$$

   where $\sigma_{Gr} = 5$ m and $\sigma_{Gv} = 0.02$ m/s.

4. Formulate the measurement innovation vector:

$$\delta \mathbf{z}_k^- = \begin{bmatrix} L_k^G - L_k^D \\ \lambda_k^G - \lambda_k^D \\ v_{N,k}^G - v_{N,k}^D \\ v_{E,k}^G - v_{E,k}^D \end{bmatrix} - \mathbf{H}_k \hat{\mathbf{x}}_k^- \tag{30}$$

   where superscript G denotes the GNSS-indicated solution from step 2 and superscript D denotes the DR-indicated solution from step 4. After updating the state estimates and error covariance matrix

using Equ 11, correct the DR solution at each epoch:

$$\begin{aligned} L_k^C &= L_k^D - \delta L_k^+ \\ \lambda_k^C &= \lambda_k^D - \delta \lambda_k^+ \\ v_{N,k}^C &= v_{N,k}^D - \delta v_{N,k}^+ \\ v_{E,k}^C &= v_{E,k}^D - \delta v_{E,k}^+ \end{aligned} \tag{31}$$

where the superscript C denotes the corrected DR solution.

Result and Discussion

1. Initializing Position and Velocity

The result of the initial position and velocity of the lawnmower is shown in Fig 2, which is iterated 5 times.

```
After 5 iterations, latitude, longitude, height, velocity and 2 offsets:
          51.5092544612951

         -0.161045484926385

           38.8258169165347

         -0.016004524923771
          0.0472025478377031
        0.00941575106054107

           10008.8009519963

           100.012351101855
```

Figure 2: Result of Initializing Position and Velocity

At $t = 0$, the latitude, longitude and height are $51.509254°, -0.161045°,$ and 38.825817m. The velocities in the north, east, and down directions are -0.016005m/s, 0.047203m/s, and 0.009416m/s, and the receiver clock offset for position and velocity are 10008.80 and 100.01, respectively.

2. Kalman Filtering for GNSS

Using the GNSS data, we implemented the Least Squares method as well as the Kalman Filtering method, both **with outlier detection**. By comparing the height result of the 2 methods, we found that the least squares implementation results in large errors, while the Kalman filter implementation is more reasonable.

As shown in Fig 3, the height results from the least squares method fluctuate greatly and simply do not model the height changes that occur when a lawnmower mows a leveled grass field: it is unlikely that a piece of grass would have a difference in height of more than 20 meters between the front and back of the lawn. Therefore in the end we did not choose the least squares method.

Fig 4 shows the height results of the Kalman filter implementation. Although the height fluctuates a lot in the beginning, after that, it quickly goes back to about 37m and makes only small fluctuations. This is intuitively more in line with reality: the altitude of this lawn is 37m, and there is a small height difference of less than 0.4m between the front and back of the lawn. So we end up using the Kalman filter method to calculate the GNSS results.

For outlier detection, it is always satellite 7 that is detected as the outlier, which is shown in Fig 5.

The horizontal position and velocity based on the Kalman Filter using GNSS data are shown in Fig 6. Starting at the lower left corner of the image, the lawnmower moves along a trajectory that passes over the lawn in columns and eventually stops at the upper right corner. The arrow in the image shows the direction of the velocity at that point.

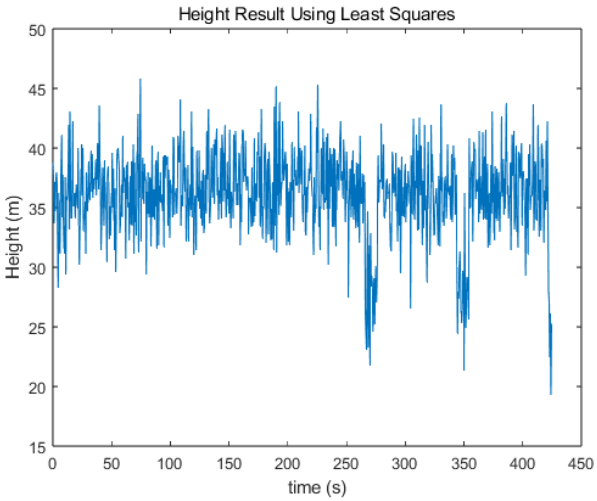

Figure 3: Height Result Using Least Squares

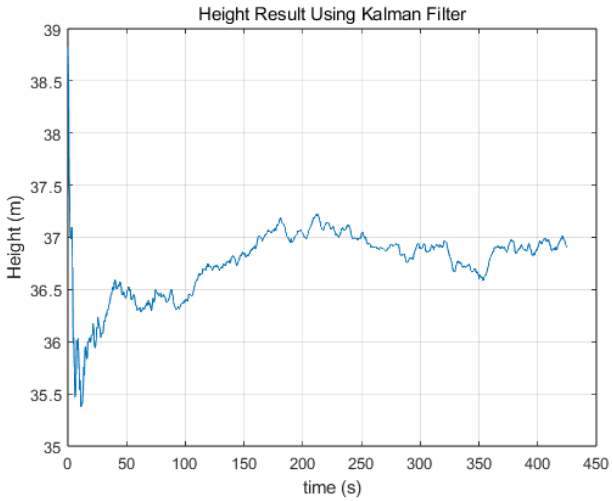

Figure 4: Height Result Using Kalman Filter

The trajectory in Fig 6 is smooth, except at the beginning. Similar to Fig 4, the initial jitter in the vector field graph is speculated to be probably due to the iterative least squares method used to initialise the position and velocity. Overall, however, the Kalman filter yields a more realistic lawnmower trajectory. And the jittery part in the lower left corner will be corrected afterward with integration with DR.

```
outlier occurred to satellite 7 at time 266
outlier occurred to satellite 7 at time 266.5
outlier occurred to satellite 7 at time 267
outlier occurred to satellite 7 at time 267.5
outlier occurred to satellite 7 at time 268
outlier occurred to satellite 7 at time 268.5
outlier occurred to satellite 7 at time 269
outlier occurred to satellite 7 at time 269.5
outlier occurred to satellite 7 at time 270
outlier occurred to satellite 7 at time 270.5
outlier occurred to satellite 7 at time 271
outlier occurred to satellite 7 at time 271.5
outlier occurred to satellite 7 at time 272
outlier occurred to satellite 7 at time 272.5
outlier occurred to satellite 7 at time 273
outlier occurred to satellite 7 at time 273.5
```

Figure 5: Outlier Detection

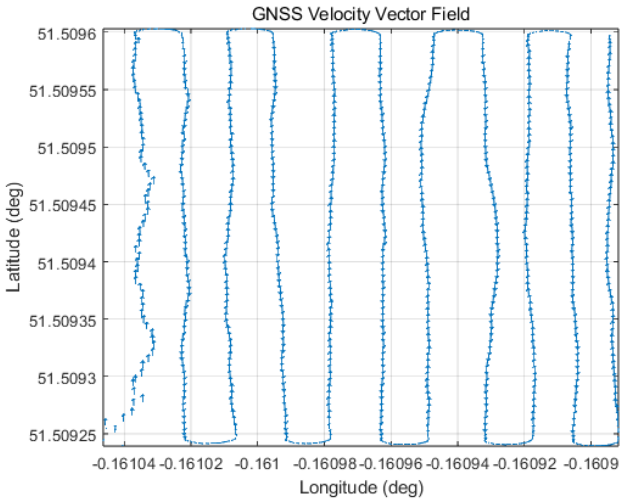

Figure 6: Pos and Vel Result Using Kalman Filter

CONTINUED

3. Kalman Filtering for Heading Correction

The heading derived by the sensors as well as the integrated heading solution are shown in Fig 7, in which the green line is the gyro-derived heading, the green line is the heading from the compass and the blue line is the fused solution.

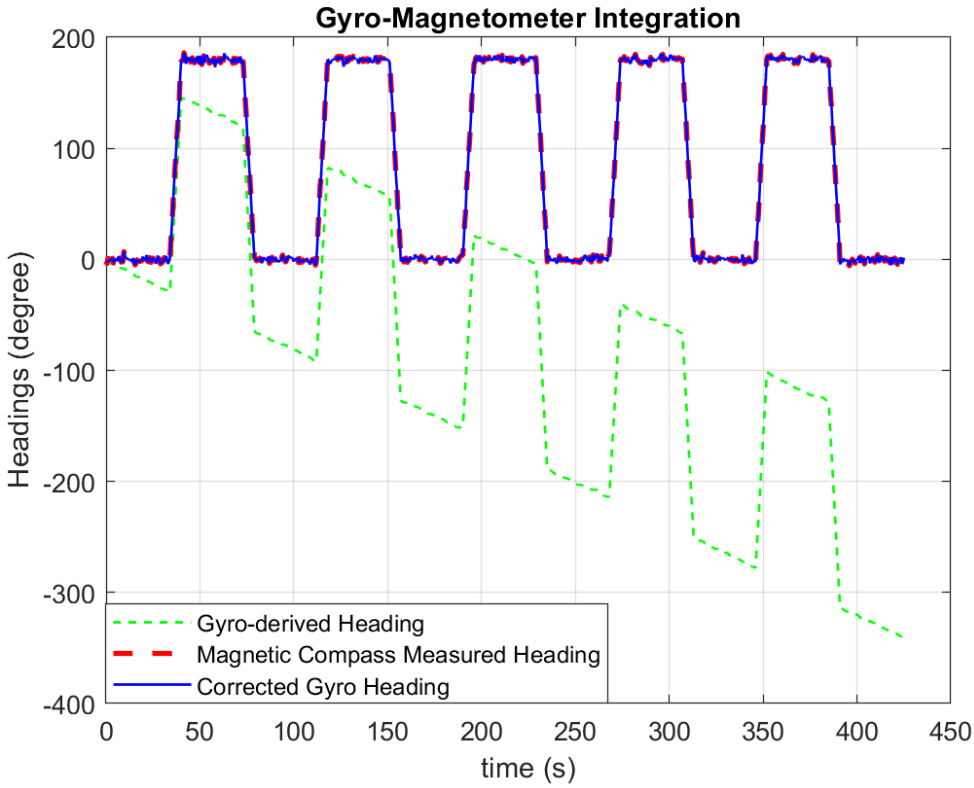

Figure 7: Result of Gyro-Magnetometer Integration

The result in the figure above shows clearly that the error of the gyro-derived heading accumulates over time, while the corrected heading closely follows the one from the magnetometer except for the few seconds at the beginning. This result matches the expectation that the long-term accuracy of the compass is kept while the unstable initial period is smoothed by the gyroscope. The coupling error was not considered since only one DOF of the gyroscope was used. The scale factor error was not considered as well, while it has been corrected during the fusion since the overall trajectory is now following the compass.

4. Lawnmower Dead Reckoning

Fig 8 shows the speed of the four wheels extracted from `Dead_reckoning.csv`. According to the trajectory shown in Fig 6 and Fig 9, the lawnmower first turns right, and the two wheels (blue and green lines in Fig 8) on the left are faster. When the lawnmower is moving forward, the rear wheels (blue and yellow lines in Fig 8) are faster since a slip could happen more often on the driving wheels. These trend of the lines justifies the identification correctness of the column representation in `Dead_reckoning.csv`: Column 2 to 5 are the speed of the front left wheel, front right wheel, rear left wheel, and rear right wheel respectively.

When the lawnmower is rotating, the speed of the wheels on the inner side of the rotation is not negative, even not zero, which means the vehicle is not rotating around its centre of geometry, leading to the inaccuracy of the velocity model. In the scenario reflected by the speed data, the d^a_{ICR} in Equ 23 would be bigger, leading to bigger v_l and consequently bigger v_a during the rotation, extending the trajectory along longitude direction. Also, the scale factor error of the speed measurements due to tyre radii and wheel slip is not considered, which could shrink the real path slightly in both latitude and longitude compared to the current solution.

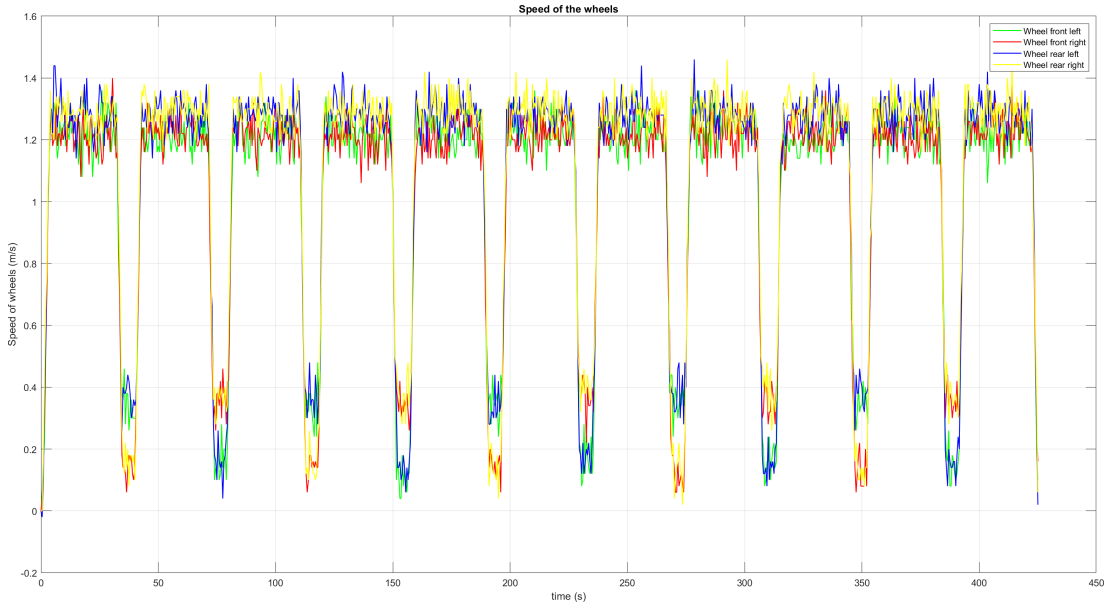

Figure 8: Speed of the wheels

The position and velocity of the lawnmower derived by Dead-reckoning are shown in Fig 9. The general trajectory resembles the one shown in Fig 6, while it goes straight up at the beginning instead of leaning to the right while leaning to the left at the end of the path.

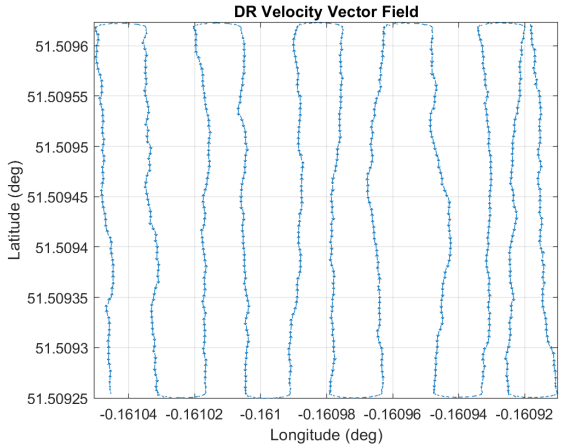

Figure 9: Pos and Vel Result Using Dead Reckoning

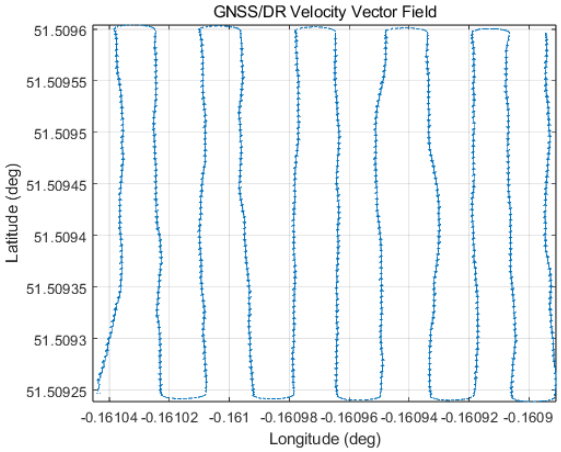

Figure 10: GNSS/DR Integration Result

5. Kalman Filtering for DR/GNSS Integration

Fig 10 shows the GNSS/DR integration result and Fig 11 shows the comparison of the 3 solutions from step 2, 4, and 5.

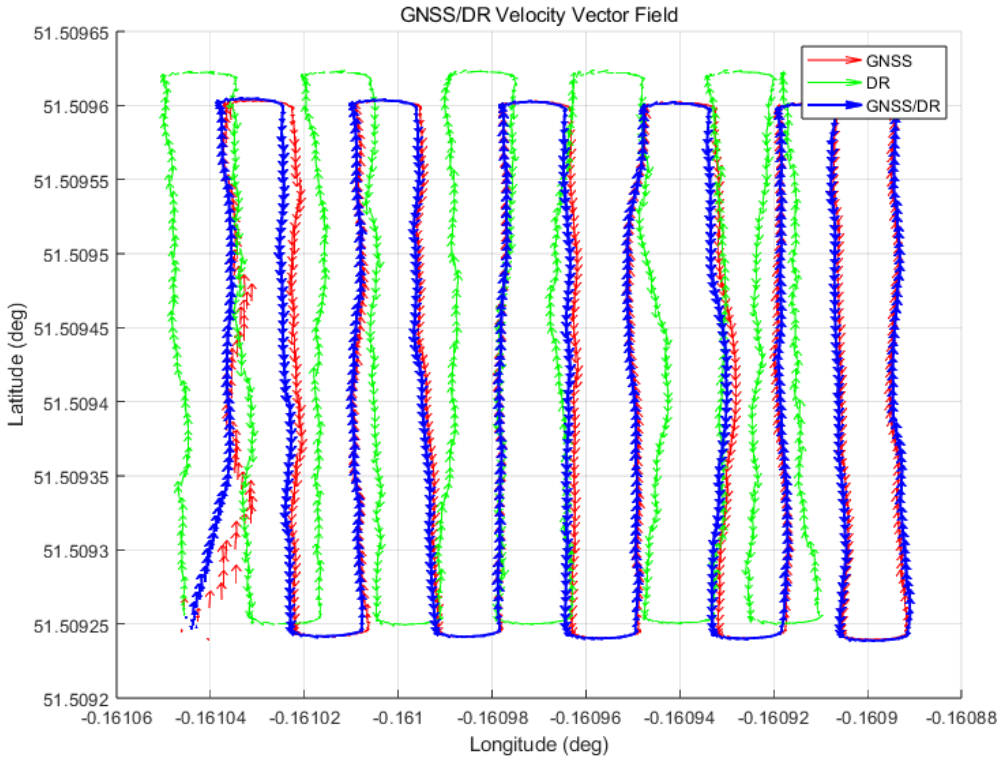

Figure 11: Comparision among the 3 Methods

Overall, the integration results are dominated by the GNSS solution, ensuring long-term accuracy. Meanwhile, taking advantage of the short-term continuity of the Dead reckoning algorithm, the trajectory is locally smoothed, especially in the lower left corner of the image, when the lawnmower starts.

It can be noted that the DR solution is more to the left compared to the GNSS or integrated results, which is a joint effect of ignoring the velocity scale factor error and inaccurate model assumption as mentioned above. Although the scale factor error, as a type of systematic error, has been calibrated with the aid of integrating the GNSS, considering more physical factors into velocity modeling could help with increasing the navigation accuracy. For instance, the speed measurements of the front wheels are not even used.

In terms of the integration method, adapting closed-loop correction to the DR system could eliminate KF DR error estimation and avoid linearisation problems, and a tightly-coupled integration could optimise the filter gain, leading to better performance. But in summary, the integration results are still better than any single GNSS or DR results, which is just the purpose of applying Kalman filtering: applying two different sensors with errors but obtaining an accurate final result with the help of Kalman gain.

References

[1] Paul Groves. Workshop 1: Mobile gnss positiong using least-squares estimation. *COMP0130: ROBOT VISION AND NAVIGATION*, 2024.

[2] Paul Groves. Workshop 2: Aircraft navigation using gnss and kalman filtering. *COMP0130: ROBOT VISION AND NAVIGATION*, 2024.

[3] Paul Groves. Multisensor integrated navigation [lecture 3b]. Lecture presented at the RVN 2021, 2021.

[4] Paul Groves. Coursework 1: Integrated navigation for a robotic lawnmower. Coursework 1 instruction at the RVN 2024, 2024.

[5] Paul Groves. Motion sensing, dead reckoning and inertial navigation [lecture 3a]. Lecture presented at the RVN 2021, 2021.

[6] Paul Groves. Workshop 3: Multisensor navigation. *COMP0130: ROBOT VISION AND NAVIGATION*, 2024.

[7] Krzysztof Kozłowski and Dariusz Pazderski. Modeling and control of a 4-wheel skid-steering mobile robot. *International Journal of Applied Mathematics and Computer Science*, 14(4):477–496, 2004.

Appendix

The code has been submitted with the report. Please check the **'cw1_group15_final_solution.m'** file in the **'code-COMP0130-cw1-Group15'** folder and run it to get all the plots and the **'Result_Group15.csv'** file. In addition to this, the folder is also uploaded to Google Drive:

https://drive.google.com/drive/folders/1vBajIybJ0gjjUlQJKgIrLJILB9obv2HU?usp=sharing

END OF COURSEWORK