

# Localisation of a Heavy-Duty Omnidirectional Vehicle Using IMU and Wheel Odometry

Xiaolong Zhang, Henri Liikanen, Eelis Peltola, Mohammad M. Aref, Jouni Mattila

Tampere University, Korkeakoulunkatu 7, 33720 Tampere, Finland

firstname.secondname@tuni.fi

**Abstract.** We introduce a localisation algorithm that uses an inertial measurement unit (IMU) and wheel odometry on a four-wheel-drive heavy vehicle for positioning. While wheel odometry alone works in simple cases without slippage, in cases that feature wheel slippage, the velocities measured by the wheel rotation show higher values. In the case of side slippage, the wheel sensors cannot observe the values. Therefore, IMUs are suitable for fusion with wheel odometry to generate real-time feedback. We use an error state Kalman filter (ESKF) to fuse the sensor information from an IMU with wheel odometry, showing results on a slow-maneuvring vehicle in tests up to five minutes in length. The IMU is an industry-grade micro-electro mechanical system (MEMS) with a gyroscope featuring 6 deg/h bias in-run stability. We use a real-time kinematic global positioning system (RTK)-GPS as a ground truth reference for the vehicle's heading angle and position. The tests results show our navigation has an accuracy of 0.3 m for position and 0.6 deg for heading angle, both within the root mean square error (RMSE) criteria. Our analysis shows that the nonlinearity of the gyroscope in the heading rotation axis is the key factor for improving performance in our implementation.

**Keywords:** IMU, KF, wheel odometry, RTK-GPS, HACV

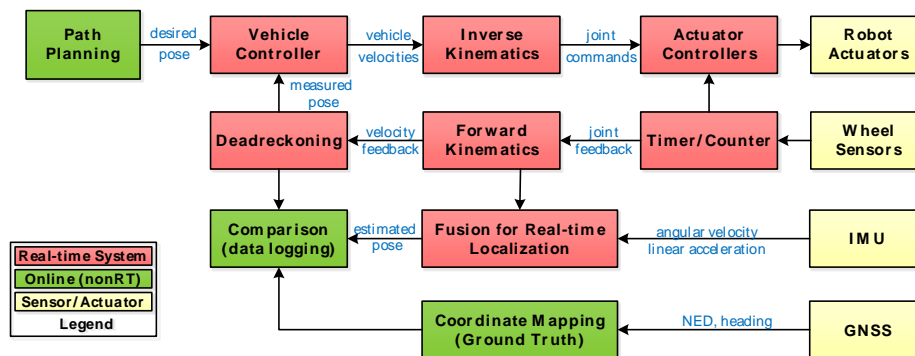
## 1 Introduction

The real-time motion control of mobile robots, such as our heavy-duty autonomous construction vehicle (HACV) test case, strongly depends on the quality of the feedback that the controller receives. Generally, sensory data for the wheels' rolling rate (i.e. wheel odometry) are the first source of information for the robot's motion. Wheel odometry is already used for the actuator-level control of each wheel, meaning these data are available for the localisation of the HACV at no hardware cost. However, using these sensors implies several practical issues.

First, due to manufacturing complexities and the harsh working environments of the robots, using fine and accurate sensors (e.g. micro pulse encoders or transducers) is not desirable. The sensor hardware must be simple, robust and easily mountable. For this purpose, metallic encoder discs are used for pulse generation with proximity sensors, such as the Hall effect or optical sensors read by counters. Therefore, we face coarse digital measurement outputs that are strongly affected by quantisation and zero-order holds. In the case of slow manoeuvring, which is common for heavy-duty vehicles, the sensors do not generate pulses on each of the control system's sample times. The output thus becomes an event-based signal as opposed to a time-based signal. For example, for a sample time of 50 ms, driving velocities below 0.25 m/s are not reliably measurable [1]. Usually, this issue causes limitations to the HACV in terms of minimum speeds for smooth feedback in autonomous driving.

Second, wheel odometry and its accuracy depend on researchers' correct assumptions about wheel-ground interaction and robot kinematics. In the case of the wheels' longitudinal slippage, the velocities measured by the wheel rotation will show higher values. In the case of the wheels' lateral slippage, the wheel sensors cannot observe the values at all. Therefore, inertial measurement units (IMUs) are a suitable choice for fusion with wheel odometry to generate real-time feedback. They are capable of transmitting absolute data at each sample time without considerable delays in operation or installation problems in manufacturing. These specifications highlight a strong need for IMUs, even when compared to global navigation satellite systems (GNSS) or radio frequency (RF)-based solutions: IMUs are not required to be in a GNSS-available environment, and they are independent of external information sources.

Based on the aforementioned issues, the main focus of this paper is on providing reliable and accurate feedback in real time by fusing wheel odometry with IMU signals. The software architecture for this is shown **Fig. 1**. We present a method that solves the minimum speed requirements for HACVs and preserves positioning accuracy that is comparable with the accuracy of an on-board GNSS.



**Fig. 1.** The software architecture designed to validate the estimation outputs by wheel odometry and IMU in comparison with ground truth (GNSS)

We use an error state Kalman filter (ESKF) [1] to fuse the sensors' information. Moreover, we remove the state of local gravity bias, as our HACV only moves in a small area, and we can set the gravity as a constant based on the latitude and height of our test location [3]. The wheel odometry provides two velocity elements that are parallel to the ground tangential plane, expressed in the HACV's body-fixed frame [4]. For the third velocity element, which is approximately vertical to the ground plane, we apply a non-holonomic constraint [5, 6, 7]. This constraint simply assumes the velocity in the vertical direction to be close to zero with some Gaussian noise, which applies to one more observation element of KF and improves estimation performance.

Normally, a fine initialisation is time-consuming for an IMU [8] if there is no external observation, such as GPS or visual aiding. For our implementation's use cases, the HACV operator expects the start phase to be as short as possible. Additionally, our IMU cannot be used to find geographic North. Our algorithm applies a coarse initialisation at the beginning phase of each test, where the HACV is kept stationary for about 15 seconds. This allows the estimation of biases for the gyroscopes and accelerometers in addition to the initial roll and pitch angles of the HACV. We define the initial heading angle to be zero, as we are mainly concerned with the relative angle for the heading. We achieve accurate estimates of the initial roll and pitch angles merely after running the stationary model. The biases of the gyroscopes along these two axes are also accurate because the estimation uses gravity as a reference.

Through simulations with the test data, we find that small disturbances for the biases of gyroscopes in roll ( $x$ -axis) and pitch ( $y$ -axis) directions do not affect the estimation results. Thus, we only apply the correction for Earth's rotation along the  $z$ -axis, which is approximately along the direction of gravity and away from the ground. We make this correction by introducing the incremental angle of the Earth's rotation around the  $z$ -axis into the heading angle observation. For the same reason, we only apply the scale factor for the gyroscope's nonlinearity to the  $z$ -axis – the scale factor of the gyroscope in this direction contributes to the main part of the error of the heading angle.

Recently, researchers have used artificial intelligence (AI) combined with traditional filters for vehicle localisation with both IMU and wheel odometry sensors [9] or with IMUs alone [10, 11]. In [9], both the propagation and measurement functions of a dynamic model are improved with neural networks and stochastic variation inference. [10] and [11] use a neural network to use an invariant extended Kalman filter (IEKF) as an adaptive filter. Both [10] and [11] manipulate the filter's covariance or observations according to the different motion patterns of their vehicles. Combining [10] and [11], it is obvious that linear acceleration in the horizontal plane can correct the accumulated error of the yaw angle. In this paper, we follow the traditional KF approach, using two kinds of patterns for HACV movement: a stationary model and a moving model. However, we do not separate a model from the moving model in which linear acceleration is introduced in the horizontal plane with no rotation of the HACV. This is suitable for our short-term navigation (i.e. within five minutes). In our future work, we would like

to experiment with using an adaptive filter for the long-term localisation of this kind of HACV.

## 2 Mathematical Background

In this section, subsection 2.1 describes the propagation for the nominal state with high-frequency IMU data and the estimated biases of the sensors. Subsections 2.2 and 2.3 include the prediction and observation models we used for the ESKF. The two observation models are for the two cases when the HACV is kept stationary and for the case when the HACV is moving. When the HACV is stationary, we use gravity as a reference for estimation, and while the HACV is moving, we use wheel odometry for the model's velocities, which are parallel to the observed motion plane. Subsection 2.4 outlines updates to the error state and the covariance matrix. For the nominal state updates, please refer to [2].

### 2.1 Nominal State Kinematics

The nominal state does not account for noise and model imperfections [2]; further, it corrects accumulated error by feeding it back to the error state. The nominal state is propagated as

$$\begin{bmatrix} p_k \\ v_k \\ q_k \end{bmatrix} = \begin{bmatrix} p_{k-1} + v_{k-1}\Delta t + \frac{1}{2}\{(R_{b\ k-1}^n f_{k-1} - g)\}\Delta t^2 \\ v_{k-1} + (R_{b\ k-1}^n f_{k-1} - g)\Delta t \\ \frac{1}{2}\Omega(\omega_{k-1}\Delta t)q_{k-1} + q_{k-1} \end{bmatrix} \quad (1),$$

where the subscript  $k$  indicates the time step.  $P$  and  $v$  are the nominal position and velocity, respectively, expressed in the navigation frame.  $q$  is the nominal orientation quaternion of the body-fixed frame with respect to the navigation frame.  $\Delta t$  is the interval of the sample time.  $g$  is the local gravity vector of the test place.  $R_b^n$  is the rotation matrix; it transfers the specific force  $f$  expressed in the body-fixed frame to express it in the navigation frame.  $R_b^n$  is a function of the orientation quaternion  $q$ .

The specific force is

$$f = a_m - a_b \quad (2),$$

where  $a_m$  is the measurement of the accelerometers and  $a_b$  is the current estimated bias of the triad-axis accelerometer.

The angular rate  $\omega$  in (1) is

$$\omega = \begin{bmatrix} \omega_{mx} & \omega_{my} & (1-k)\omega_{mz} \end{bmatrix}^T - \omega_b \quad (3),$$

where  $\omega_{mx}$ ,  $\omega_{my}$  and  $\omega_{mz}$  are the measurements of the gyroscope in x, y and z directions, respectively;  $\omega_b$  is its current estimated angular drift;  $k$  is the scale factor, which is a constant value indicating the non-linearity of the gyroscope. We only apply the scale factor in the z direction of the gyroscope.

In (1),  $\Omega$  is a function of the angular rate and sample interval, taking the form

$$\Omega(\omega\Delta t) = \Delta t \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix} \quad (4)$$

On each time step, we normalise the orientation quaternion  $q_k$ .

## 2.2 Prediction Step of KF

The state has 15 elements:

$$\delta x = [\delta p \quad \delta v \quad \delta \theta \quad a_b \quad \omega_b]^T \quad (5),$$

where  $\delta p$  is the position error and  $\delta v$  is the velocity error, respectively.  $\delta \theta$  is the error of orientation in Euler form, with the three elements being roll, pitch and yaw;  $a_b$  and  $\omega_b$  are the biases of the accelerometers and the gyroscope, respectively. Each time there is an update in the error state of KF, we reset the error states of position, velocity and orientation as zero; thus, in (16), we ignore the error state. Since the biases  $a_b$  and  $\omega_b$  have relatively small values, in practice, we keep their state values instead of resetting them to zero.

In KF, we use the Euler form to represent the orientation error; however, this is in quaternion form for the state propagated in (1). Our IMU's sample frequency is approximately 1000 Hz; propagating the system state (5) requires the transformation matrix

$R_b^n$ . This process only involves algebraic operations and does not include sine or cosine functions. The observation datum for the error state KF is 10 Hz – as the KF updates the state, the attitude must be transferred from the quaternion form to the Euler angle form and then transferred back again. This involves sine and cosine functions; however, it only runs at 10 HZ, which is much lower than 1000 HZ. So the computation load is not a problem, even when we apply a relatively higher sample frequency of 1000 HZ.

The transition matrix is

$$F = \begin{bmatrix} I & I\Delta t & O & O & O \\ O & I & -[R_b^n(a_m - a_b)]_x \Delta t & -R\Delta t & O \\ O & O & I & O & -R\Delta t \\ O & O & O & I & O \\ O & O & O & O & I \end{bmatrix} \quad (6),$$

where  $I$  is a 3 x 3 identity matrix and  $O$  is a 3 x 3 zero matrix.  $[R_b^n(a_m - a_b)]_x$  is the accelerometer output minus its bias, rotated into the navigation frame and expressed in a skew-symmetric form.

The state covariance propagate as

$$P_{k+1}^- = F_k P_k F_k^T + B Q_i B^T, \quad (7),$$

where the covariance matrix of perturbation impulses is

$$Q_i = \begin{bmatrix} \sigma_a^2 \Delta t^2 I & O & O & O \\ O & \sigma_w^2 \Delta t^2 I & O & O \\ O & O & \sigma_{ar}^2 \Delta t I & O \\ O & O & O & \sigma_{\omega r}^2 \Delta t I \end{bmatrix} \quad (8)$$

In (8),  $\sigma_a^2$  and  $\sigma_w^2$  are the covariance for the measurement noise of accelerometer and gyroscope, respectively.  $\sigma_{ar}^2$  and  $\sigma_{\omega r}^2$  are the covariance for the random walk of accelerometer's bias and gyroscope's drift, respectively.

The Jacobian of the perturbation in (7) is

$$B = \begin{bmatrix} O & O & O & O \\ I & O & O & O \\ O & I & O & O \\ O & O & I & O \\ O & O & O & I \end{bmatrix} \quad (9)$$

### 2.3 Observation Step of KF

When the HACV is stationary, we take the velocity to be zero in three directions and fix the heading angle as an initial value or the value right before the HACV's velocity becomes zero minus the current estimated velocity and current estimated heading:

$$z_{k+1} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \psi_{fix} - \omega_{earth} \sin(La)t_{k+1} \end{bmatrix} - \begin{bmatrix} \hat{v}_x \\ \hat{v}_y \\ \hat{v}_z \\ \hat{\psi} \end{bmatrix} \quad (10)$$

Here, the current estimate is the best-estimated states of the previous time step propagated forward one time step using (1).  $\psi_{fix}$  is the fixed value for the heading angle when the HACV is stationary;  $\omega_{earth}$  is the Earth's rotation rate;  $La$  is 61.4°N, the latitude for our test location;  $t_{k+1}$  is the current time from the test's beginning.

The Jacobian of (10) is

$$H_{k+1} = \begin{bmatrix} 0_{3 \times 3} & I_{3 \times 3} & 0_{3 \times 9} \\ 0_{1 \times 8} & 1 & 0_{1 \times 6} \end{bmatrix} \quad (11)$$

When the HACV is moving, the observation is

$$z_{k+1} = fC_o^T \begin{bmatrix} v_{bx} \\ v_{by} \\ 0 \end{bmatrix} - V_b \quad (12),$$

where  $v_{bx}$  and  $v_{by}$  are the velocities of the HACV's centre of control in x and y directions, expressed in the body-fixed frame of the wheel odometers and measured by

the wheel odometers. We set the velocity of the HACV in z direction to zero using the non-holonomic constraint [6, 7].  $f$  indicates the efficiency of the wheel odometer; since we carefully calibrated our system, we set the value as one for the test in this paper.  $C_o^I$  is the matrix denoting any misalignment between the velocity of the wheel odometer and the body-fixed frame of the HACV, which we set as identical for our tests.  $V_b$  is the current estimated HACV velocity, expressed in the body-fixed frame of the HACV. It is a function of the estimated velocity  $\hat{v}_n$  and the attitude transformation matrix from the navigation frame to the body-fixed frame  $R_n^b$ .

$$V_b = R_{nk+1}^b \hat{v}_n - \omega \times l \quad (13),$$

where  $\omega$  is defined in (3) and  $l$  is the coordinate of the IMU in the body-fixed frame of the HACV, defined as the constant vector  $l = [-1.18 \ 0 \ 0.4]^T$  m for our case.

The Jacobian of (12) is

$$H_{k+1} = \begin{bmatrix} 0_{3 \times 3} & R_{bk+1}^n & 0_{3 \times 6} & -L \end{bmatrix}, \quad (14),$$

where  $L$  is the skew-symmetric form of coordinate vector  $l$ .

## 2.4 Update Step of KF

The gain of the KF,  $K$ , is expressed as

$$K = P_{k+1}^- H_{k+1}^T (H_{k+1} P_{k+1}^- H_{k+1}^T + V)^{-1}, \quad (15),$$

where  $V$  is the matrix of observation noise (depending on which model we use): the stationary model from (10) and (11) or the moving model from (12) and (14).

The error state of each time step is

$$\delta x = K z_{k+1} \quad (16)$$

The update for the covariance matrix is

$$P_{k+1} = (I_{15 \times 15} - KH) P_{k+1}^- \quad (17)$$

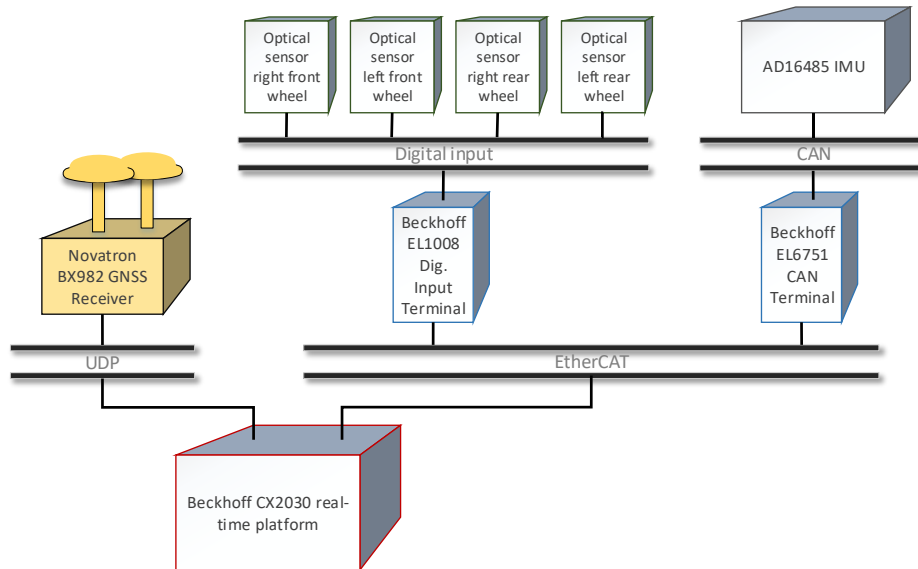
Once the observation data come in, the KF outputs the error state  $\delta x$ ; this error state is fed back to the system propagator as defined in (1). Then, the system state, position, velocity and orientation are updated as detailed in [1].

### 3 Experiment Platform

We conducted our experiments with a Haulotte 16RTJ PRO heavy-duty articulating lift boom with four-wheel drive and active steering [12]. For autonomous driving, we fitted this HACV with a Beckhoff CX2030 real-time platform, which has several modules for interfacing with different sensors and actuators. For each wheel, we installed a Panasonic PM photoelectric odometry sensor, sensing 150 equally distributed holes in a metallic disc; they have a data rate of 1000 Hz.

The IMU used in the test, installed on the body of the HACV, is an Analog Devices ADIS16485 MEMS IMU. Its in-run stabilities are 6 deg/h angular for its triaxial gyroscope and 32  $\mu\text{g}$  for its triaxial accelerometer. It is connected to the CX platform with a controller area network (CAN).

We used a Novatron BX982 differential carrier-phase GNSS receiver with two Trimble LV59 antennas to provide a reference for positioning. The antennas are located 2.6 meters apart on the lift boom HACV; one antenna gets the position of the HACV, and the other is used to form the heading vector for the HACV yaw angle with the first antenna. A commercial base station located 13 kilometres away gives an RTK reference point for correction. Using this correction, the accuracy for horizontal position is  $\pm 0.02$  meters, and the accuracy for the yaw angle is  $\pm 0.09$  degrees. The GNSS receiver has a data rate of 50 Hz and is connected to the CX platform via Ethernet user datagram protocol (UDP).



**Fig. 2.** Project-specific sensor architecture on the Haulotte

**Fig. 2** shows the sensors and their connections to the real-time platform. The GNSS receiver also receives its RTK corrections via a mobile network connection on a router in the lift boom HACV.

The Beckhoff CX platform runs the TwinCAT software system; it reads all the sensors. Underlying the sensor connections shown in **Fig. 2**, the CX platform communicates with its modules in EtherCAT. With the exception of the UDP interface, which we wrote in the TwinCAT PLC language, we did all of our programming in Matlab Simulink and built for the TwinCAT system from Simulink.

We performed the tests in Finnish winter conditions, with some sleet on the ground resulting in wheel slippage. The lift boom HACV is shown in **Fig. 3** in the test conditions. A path-following controller handled the steering of the HACV with no manual manoeuvring. We conducted several runs ranging from 1–5 minutes using planned paths.



**Fig. 3.** The Haulotte lift boom in test conditions

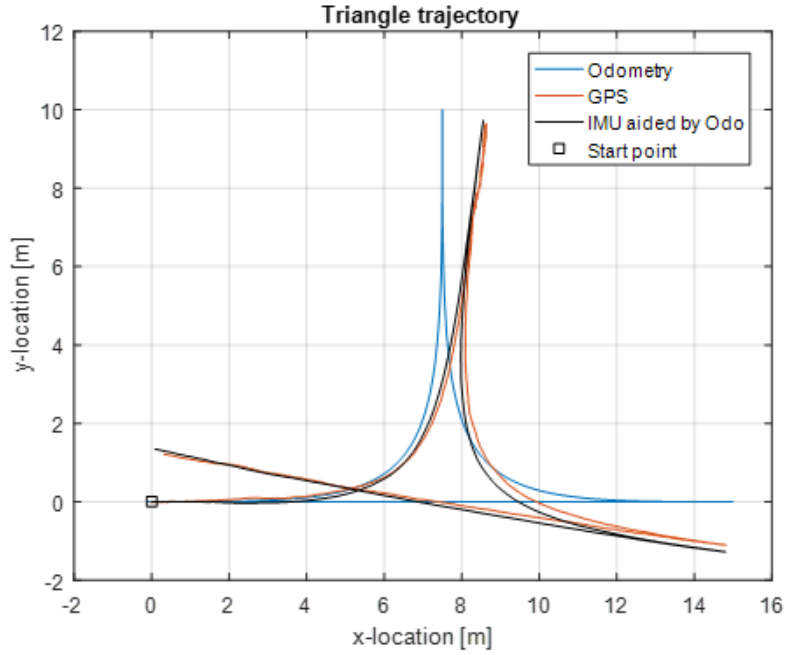
## 4 Test Results

We include test cases in this article: one trajectory is roughly a triangle shape, and the other is roughly a square shape. Because the installation position of the IMU is at an offset with respect to the control centre of the wheel odometry, the position outputs from KF are transferred to the frame that the wheel odometry uses, represented as a black line in **Fig. 4.** and **Fig. 6.** Similarly, the position outputs from RTK-GPS are also transferred to the centre of the wheel odometer and used as ground truth. They are represented as a red line; the blue line is the HACV's position trajectory calculated only using wheel odometry.

The yaw angles, or HACV heading's evolution in time, are shown in **Fig. 5.** and **Fig. 7.** for the two test cases. As in **Fig. 4.** and **Fig. 6.**, the red line is from the RTK-GPS and is regarded as the ground truth; the blue line represents the wheel-odometry-only calculation. The black line is the yaw estimated by KF. The initial heading angle is set as zero, and the yaw angle output is the angle from the GPS minus its initial value.

During the tests, we kept the engine on at all times. We kept the HACV stationary at the initial stage; KF uses (10) for observation. When the HACV started moving, the observation of KF switched to (12) and remained as such until the test was complete. For the square shape movement, from 0 s to 18 s, and from 155 s to 185 s, we kept the

HACV stationary. For the triangle movement, the stationary phases are from 0 s to 15 s and from 255 s to the test's end.

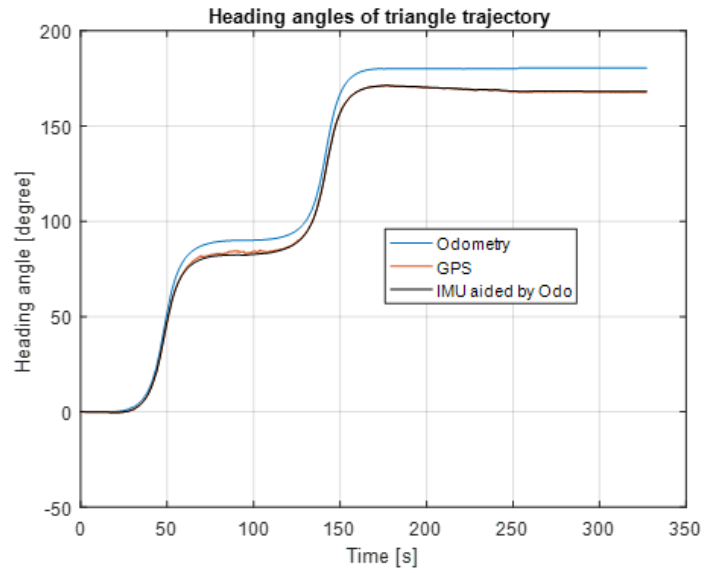


**Fig. 4.** The trajectory estimated by IMU and wheel odometry roughly matches the ground truth from RTK-GPS. Adding the IMU compensates for the effect of slippage that is apparent when only using the wheel odometry.

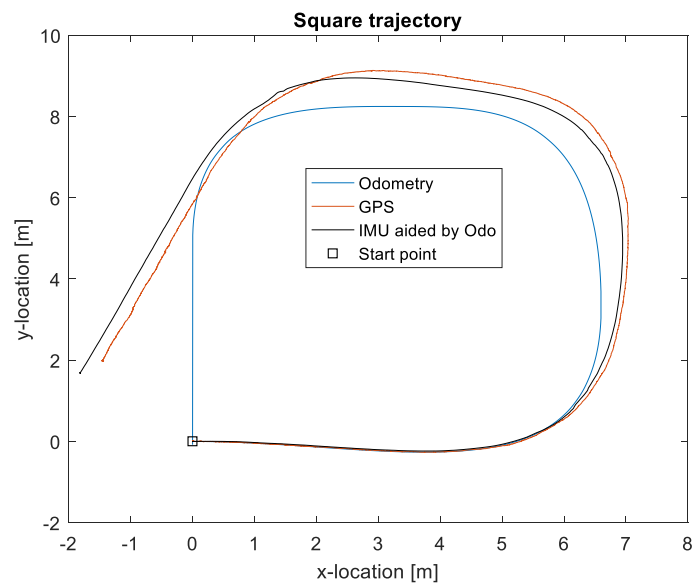
**Table 1.** Test results

Trajectory	Position RMSE (m)	Heading RMSE (Degree)	Maximum Position Error (m)	Maximum Heading Error (Degree)
Triangle	0.199	0.55	0.310	2.27
Square	0.314	0.56	0.490	2.28

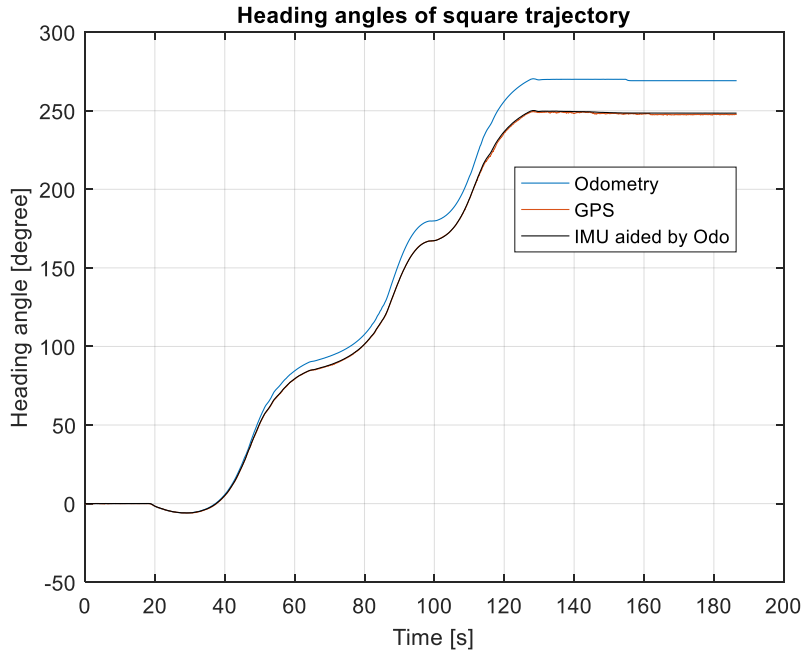
**Table 1** summarises the results of the two test cases. The RMSE of the position represents the absolute distance error in the x–y plane compared with the RTK-GPS output. The output frequency of RTK-GPS is 50 Hz, and the 1000 Hz output of KF is sampled down to 50 Hz when calculating the test results.



**Fig. 5.** The heading angle of the HACV continually increases from 0 deg to about 180 deg. Combined with the IMU, the estimated results nearly match the ground truth from the RTK-GPS.



**Fig. 6.** The trajectory generated by wheel odometry only drifts away compared to the ground truth; however, the trajectory of IMU aided by wheel odometry roughly matches the ground truth.



**Fig. 7.** The heading angle of the HACV continually increases from 0 deg to about 250 deg. Combined with the IMU, the estimated result nearly matches the ground truth from the RTK-GPS. However, using wheel odometry only, the error of the heading angle deviates from the ground truth after each sharp turn of the HACV.

## 5 Discussion

At the beginning of each test, we keep the HACV stationary for 10–20 seconds to determine the biases of gyroscope and accelerometer and to correct the initial roll and pitch angles. The initial yaw angle we simply set as zero.

The in-run stability of the IMU’s gyroscope is 6 deg/h [13]; this accuracy is not enough to justify using only IMU to determine geographical direction. However, for a period of several minutes, the drift of the gyroscope’s bias should be small. Moreover, the HACV is moving on ground covered with ice and snow – a very slippery environment – which means the observation velocities from wheel odometry,  $v_{bx}$  and  $v_{by}$ , have high deviations from the true velocities of the HACV base. Using them to correct

the altitude of the HACV and the IMU biases is difficult. Because slipping happens frequently, if we tried to use the wheel odometry to correct the attitude, the output of the HACV velocity would deform. In [8], the authors mention that the added linear acceleration helped them to resolve the ambiguity of IMU bias. However, for our implementation, when there is linear acceleration when moving, slipping occurs regularly; thus, we do not use angular velocity from the wheel odometry in the observation step of the KF.

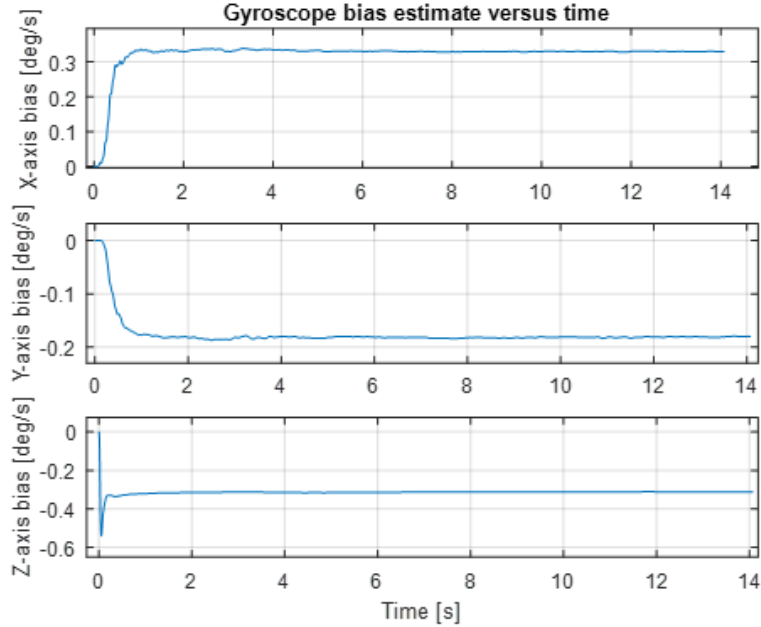
Now, we summarise the strategy of tuning the parameters of the KF: during the HACV's stationary model, we let the KF converge as quickly as possible to get the biases  $a_b$  and  $\omega_b$ . When switching to the moving model, we update the state using the observations  $v_{bx}$  and  $v_{by}$  by changing the velocity and not the attitude. The parameters for the KF process noise in (8) are summarised in **Table 2**.

**Table 2.** Process noise

$\sigma_{\bar{a}}$	$\sigma_{\bar{\omega}}$	$\sigma_{ar}$	$\sigma_{\omega r}$
0.05 m/s	5 deg/s	0.007 m/s <sup>1.5</sup>	0.0001 deg/s <sup>0.5</sup>

We set the standard deviations for observation noise in the stationary model at 0.02 deg/s for velocity and 0.05 deg/s for heading angle. For the moving model, we set the noise parameter for velocity observation at 7 deg/s and the noise parameter for non-holonomic velocity at 2.5 m/s.

We set the initial standard deviation for the gyroscopes' bias at 0.8 deg/s and at 0.02 m/s<sup>2</sup> for the accelerometers' bias.



**Fig. 8.** As the HACV is kept stationary, the bias of the gyroscope converges to a steady state within several seconds. We can use the steady state values for short-term navigation even without updating the state of bias.

As an example, **Fig. 8.** shows the output of the estimated bias for the gyroscope in the triangular trajectory test when the KF is in the stationary model. The gyroscope's bias roughly converges to a steady state within several seconds in all three directions. When the KF switches to the motion model after 14 s, these biases keep a roughly constant value. In the x and y directions, the estimation of the gyroscope's biases can withstand an error of at least up to 0.02 deg/s. With the final values of the gyroscope's bias just after the stationary model starts running, we only add disturbances in the x and y directions, the standard deviation of which is 0.02 deg/s. We inject these three gyroscope biases as the initial state for the KF and set the initial uncertainty of the gyroscope's bias as a very small value: 0.0001 deg/s. We find that the output results have no significant difference compared to when the gyroscope's initial biases are set to zero: only a 1 or 2 cm position RMSE, less than 0.2 deg in yaw angle RMSE. This is because in terms of the roll and pitch directions, the angle estimation can use gravity as a reference.

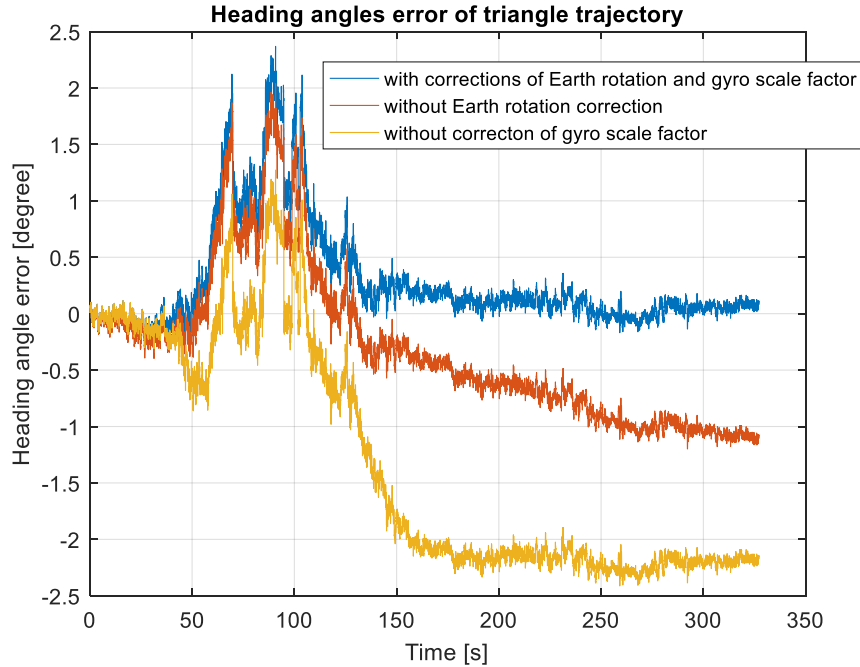
However, in the z direction (i.e. mainly for the yaw rotation), the results are very sensitive to the gyroscope's bias. Running a motion model for 5 min, we expect the estimation accuracy of the gyroscope's bias in the z direction to reach 0.002 deg/s just after the stationary model stops running. Because there is no other information except

for wheel odometry, and because the KF does not use the information from the wheel to correct the attitude (as mentioned previously), the yaw angle accuracy mainly depends on the accuracy of the estimated angular rate in the z direction.

In (10), we add the Earth's rotation effect only for the yaw angle observation. Because the gyroscope's z direction is roughly aligned with gravity, the maximum roll and pitch angles are less than 2 deg. As the HACV only moves in a small area with a known latitude, we set the latitude to a constant value in the test. Thus, the Earth's rotation vector projected into the z direction of the gyroscope is roughly constant. Projecting the Earth's rotation into the x and y axes of the gyroscope results in small values, and they do not affect the test results even when completely ignored in our model. Again, this is because we have gravity as a reference for the roll and pitch angles.

**Fig. 9.** shows the heading angle error, which we generate by comparing RTK-GPS and KF output with three different settings: the red line corresponds to removing the Earth's rotation in (10). Compared with the blue line, where no removal was done, we notice that for motion taking place for longer than 300 s, the accuracy of the heading angle is about 1 deg worse without the correction of the Earth's rotation. In (10), the value of  $\psi_{fix}$  is in respect to an Earth-fixed frame; we also observe it in the Earth-fixed frame. Then, the value for  $\psi_{fix} - \omega_{earth} \sin(La)t_{k+1}$  is with respect to the inertial frame but observed in the Earth-fixed frame for the still HACV; the strap-down gyroscope measures the angular rate with respect to the inertial frame. Therefore, the stationary model of our KF outputs a gyroscope's bias in the z-axis, which roughly removes the effect of the Earth's rotation.

From 15 s to 175 s in **Fig. 9.**, the HACV rotates with an angular rate with a maximum value less than 5.2 deg/s in the yaw direction. The yellow line shows that without the scale factor correction, the heading angle error increases more than 2 deg. The scale factor only has an effect when there is an angular rate. For example, from 175 s to 275 s, the HACV moves in a straight line and then remains still until the end. Therefore, after 175 s, the error of the heading angle does not increase even without correcting the scale factor.

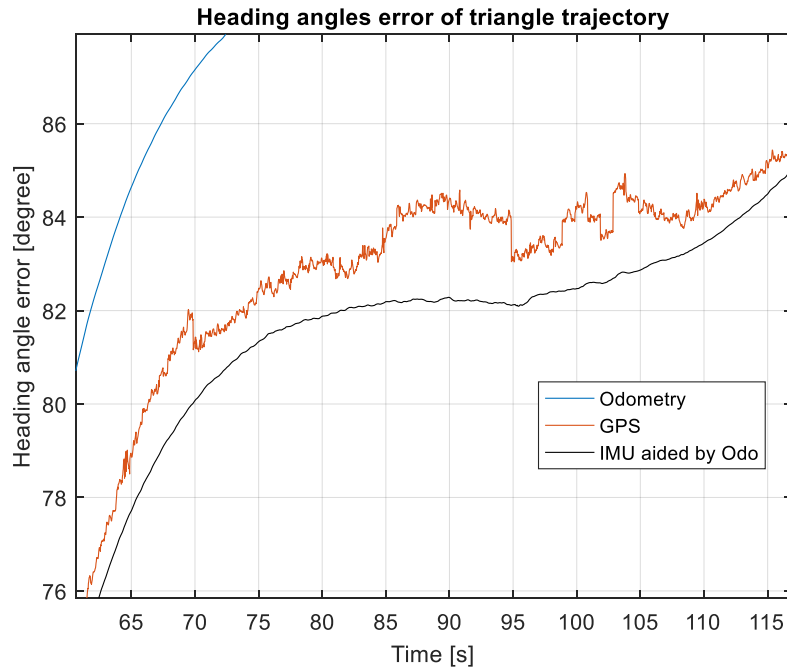


**Fig. 9.** The result of the estimation for yaw angle improves significantly by introducing the Earth’s rotation into the yaw angle observation and including a scale factor in the z-axis for the gyroscope.

For both the triangle and square trajectory tests, the scale factor in (3) takes a constant value of 0.013 after intensive simulations and trials; both of these tests with this scale factor yield optimal results. The scale factor can be regarded as a local parameter; for our tests, the maximum angular rate is less than 6 deg/s, but the full measurement range for the gyroscope is 450 deg/s. The gyroscope’s scale factor mainly depends on temperature and the input angular rate, and it can be written as a polynomial formula with these two parameters [14]. For our implementation, we only needed to do a pre-calibration for the gyroscope’s bias in the z-axis within a range of angular rate and a range of temperature.

As mentioned previously, our KF does not use the velocity from wheel odometry to correct the altitude states, as it mainly constrains the magnitude of the velocity states. Additionally, our gyroscope’s measurement and bias has no jumps. However, in **Fig. 9.**, from about 45 s to 140 s, the error of the heading angle has several peaks. This is because our RTK-GPS has some abnormalities for the yaw angle output during that time. In **Fig. 10.**, we enlarge the local part of **Fig. 5.**, clearly showing that the heading

angle from GNSS has some jumps, the magnitudes of which reach 1 deg. This phenomenon is shown in the square trajectory test. This also indicates that the maximum error of the heading angle may in reality be smaller than 1 deg for our tests.



**Fig. 10.** The RTK-GPS output for the yaw angle's reference has abnormalities in several points.

## 6 Conclusion

For an HACV that moves slowly on roughly level ground, we use a high-end industry-grade MEMS IMU aided with wheel odometry to obtain an accuracy of heading angle within 1 deg (and 50 cm accuracy for position). The navigation time span is about 5 min, using a coarse initial alignment of less than 15 seconds of data when the HACV is kept stationary. The main limitation is the nonlinearity of the gyroscope on the heading rotation axis; delicate pre-calibration of the scale factor for this axis of the gyroscope would significantly improve system performance. For our implementation, we only needed to carry out the pre-calibration of the scale factor within a small angular rate range and a specific temperature range.

## References

1. Aref MM, Ghabcheloo R, Mattila J (2014). A macro-micro controller for pallet picking by an articulated-frame-steering hydraulic mobile machine. In: 2014 IEEE International Conference on Robotics and Automation (ICRA). IEEE Press, New York, pp. 6816–6822. doi: 10.1109/ICRA.2014.6907865
2. Sola J (2017) Quaternion kinematics for the error-state Kalman filter. arXiv preprint arXiv:1711.02508.
3. Noureldin A, Karamat TB, Georgy J (2012) Fundamentals of inertial navigation, satellite-based positioning and their integration. Springer Science+Business Media.
4. Aref MM, Oftadeh R, Ghabcheloo R, Mattila J (2015) Fault tolerant control architecture design for mobile manipulation in scientific facilities. *International Journal of Advanced Robotic Systems* 12(1):4.
5. Shin E-H (2001) Accuracy improvement of low cost INS/GPS for land applications. University of Calgary.
6. Dissanayake G et al. (2001) The aiding of a low-cost strapdown inertial measurement unit using vehicle model constraints for land vehicle applications. *IEEE transactions on robotics and automation* 17.5: 731–747.
7. Wu Y. (2014) Versatile land navigation using inertial sensors and odometry: Self-calibration, in-motion alignment and positioning. In: 2014 DGON Inertial Sensors and Systems (ISS). New York, IEEE Press.
8. Wu Y, Zhang H, Wu M, Hu X, Hu D (2012) Observability of strapdown INS alignment: A global perspective. *IEEE Transactions on Aerospace and Electronic Systems* 48(1): 78–102.
9. Brossard M, Silvere B (2019). Learning wheel odometry and IMU errors for localization. In: International Conference on Robotics and Automation (ICRA), May 2019, Montreal, Canada.
10. Brossard M, Barrau A, Bonnabel S (2019) RINS-W: Robust inertial navigation system on wheels. arXiv preprint arXiv:1903.02210.
11. Brossard M, Barrau A, Bonnabel S (2019) AI-IMU dead-reckoning. arXiv preprint arXiv:1904.06064.
12. Liikanen H, Aref MM, Oftadeh R, Matilla J (2019) Path-following controller for 4WDs hydraulic heavy-duty field robots with nonlinear internal dynamics. In: 10th IFAC Symposium on Intelligent Autonomous vehicles (IAV), Gdansk, Poland, July 2019. Accepted for publication.
13. Analog Devices, ADIS16485 datasheet, rev. H. <https://www.analog.com/media/en/technical-documentation/data-sheets/ADIS16485.pdf>
14. Liu CZ, Wang X, Tang QJ (2014, November). Error analysis and compensation research of scale factor for MEMS gyroscope. In: International Symposium on Optoelectronic Technology and Application 2014: Image Processing and Pattern Recognition, vol 9301, p. 93010R. International Society for Optics and Photonics.