

Article

LiDAR-Stabilised GNSS-IMU Platform Pose Tracking

Timothy D'Adamo , Tyson Phillips  and Peter McAree 

School of Mechanical and Mining Engineering, The University of Queensland, Brisbane, QLD 4072, Australia; t.phillips1@uq.edu.au (T.P.); p.mcaree@uq.edu.au (P.M.)

* Correspondence: t.dadamo@uq.edu.au

Abstract: The requirement to estimate the six degree-of-freedom pose of a moving platform frequently arises in automation applications. It is common to estimate platform pose by the fusion of global navigation satellite systems (GNSS) measurements and translational acceleration and rotational rate measurements from an inertial measurement unit (IMU). This paper considers a specific situation where two GNSS receivers and one IMU are used and gives the full formulation of a Kalman filter-based estimator to do this. A limitation in using this sensor set is the difficulty of obtaining accurate estimates of the degree of freedom corresponding to rotation about the line passing through the two GNSS receiver antenna centres. The GNSS-aided IMU formulation is extended to incorporate LiDAR measurements in both known and unknown environments to stabilise this degree of freedom. The performance of the pose estimator is established by comparing expected LiDAR range measurements with actual range measurements. Distributions of the terrain point-to-model error are shown to improve from 0.27 m mean error to 0.06 m when the GNSS-aided IMU estimator is augmented with LiDAR measurements. This precision is marginally degraded to 0.14 m when the pose estimator is operated in an *a priori* unknown environment.

Keywords: GNSS; IMU; LiDAR; perception; pose estimation; navigation system; terrain mapping



Citation: D'Adamo, T.; Phillips, T.; McAree, P. LiDAR-Stabilised GNSS-IMU Platform Pose Tracking. *Sensors* **2022**, *22*, 2248. <https://doi.org/10.3390/s22062248>

Academic Editor: Ali Khenchaf

Received: 8 February 2022

Accepted: 11 March 2022

Published: 14 March 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

An inertial measurement unit (IMU) aided by global navigation satellite system (GNSS) information is often used to track the six degree of freedom (DOF) pose (here, pose refers to the six degree of freedom position (x, y, z) and orientation (θ, ϕ, ψ) of the platform) of a moving platform. Solutions come in many forms but are broadly categorised as being tightly or loosely coupled depending on the level of sensor integration. The rationale for using the two sensor types is that GNSS receivers provide low-frequency information while the IMU provides information about higher frequency motion, noting that when stationary, the IMU measures gravity which can be considered as zero-frequency (DC) information. In a typical implementation, measurements from the two sensors are combined using optimal estimation methods, *vis-à-vis* Kalman filtering, to maintain an estimate of the pose of the platform frame P as it moves relative to a global frame G .

Where three GNSS receivers with non-collinear antenna are used, the low-frequency pose of the platform can be fully determined to the precision and bandwidth of the receivers. Each GNSS receiver provides positional knowledge of its antenna centres to a precision of approximately 0.05 m at a bandwidth from 0 Hz to 2–5 Hz (with measurement updates typically provided at 10 Hz). Higher frequency motion is then determined from information provided by the IMU, which will typically provide acceleration and angular rate information from 0 Hz to 50 Hz, at update rates of 50 Hz to 100 Hz. Such an arrangement provides a nice division of labour, with the Kalman filter-based pose estimator serving to combine the disparate frequency information provided by the two sensor types.

However, where two GNSS receivers are used, only five of the six degrees of freedom of the platform can be established using the information provided by the GNSS receivers. The platform's rotational orientation about the line drawn through the antenna centres is

not defined. In theory, gravity can be used to resolve this, provided the placement of the IMU frame does not result in the vector emanating from the sensor origin and directed by gravity, intersecting the antenna centre line. In practice, however, for a similarly structured estimator, start-up bias and the conflation of gravity measurement with accelerations due to platform motion limits the accuracy of pose estimates.

This paper develops and evaluates a method for using three-dimensional LiDAR measurements to stabilise pose estimates for platforms having two GNSS receivers. We consider two situations: (i) where the environment around the platform is *a priori* known, and (ii) where it is not. LiDAR sensors are often present on robotic platforms as components of a perception system. Their use for stabilising pose estimates becomes relevant where three GNSS receivers are impractical or cost-prohibitive.

Pose estimation is clearly ground for fertile research and has been for some time. Google Scholar identifies no fewer than 4650 publications that contain all of the terms “GNSS”, “IMU”, and “loosely coupled”. A total of 851 of these were published after 2020; the first was published in 1968 [1]. If the term “pose” is included, 1050 papers are identified; 279 of these were published after 2020, while the earliest is [2]. Adding the term “LiDAR” reduces the set to 452, of which 167 have been published since 2020, with the earliest again dating from 1981 [3]. These, and the many other papers that use alternative but equivalent terminology, span a variety of approaches and applications, from terrestrial [4–6], to aeronautical [7–11], space [12–15]. In some instances, the search terms appear in reference to the technology being used for a specific application, e.g., precision agriculture [16–19] and construction [20,21]. In others, reference is made to implementation and algorithms. These are described at various levels of detail, but often with ambiguity.

Methods for stabilising pose estimates with LiDAR measurements exist and generally fall into two categories. The first are *feature-based* methods which seek to identify features, sometimes known as landmarks, in LiDAR point clouds and maintain an estimate of their location. Features can be lines or planes [22–24], curves [25,26], or corners [27]. Feature-based methods can also match sequential LiDAR scans, solving for the incremental change in pose between scans [28–32].

An alternative approach uses *model-based* methods. These generally compare measurements from a ranging sensor with a model of the environment, searching for a pose solution that provides an optimal solution.

Ref. [33] propose an airborne solution for aiding a navigation system through the use of a large-scale terrain model and LiDAR sensors. Similarly, Ref. [34] augment pose estimates by fitting LiDAR measurements against a known two-dimensional environment model. Refs. [35,36] show how pose estimates of known objects can be obtained from only LiDAR scans by finding the pose that is most likely among the set of all possible poses.

Other work has focused on the augmentation of pose when GNSS measurements are degraded or unavailable, such as loss of real-time kinematic (RTK) corrections. Ref. [37] develop a solution that uses a known geometric model of the world and that augments GNSS measurements with a pseudo-measurement of the expected height of the receiver. Ref. [38] use a conceptually similar approach, but apply it to aerial vehicles.

Work on *model-based* methods has also focused on building an environment model simultaneously with localisation within this model. One solution, presented by [39], develops an algorithm for a LiDAR- and GNSS-aided IMU navigation system for use in an airborne platform that only maps the terrain when GNSS and IMU measurements are available and reliable. When the integrity of the GNSS-aided IMU navigation system is compromised, LiDAR observations are fitted to the generated terrain model through an ICP-based method. The authors find that the solution is robust and improves the accuracy of the pose when the GNSS system is in a degraded state.

Ref. [40] present a method that augments a GNSS-aided IMU and odometry navigation system with a scanning LiDAR sensor. The authors estimate the environment using the LiDAR scanner as they move through an urban scene, and remove dynamic obstacles by compressing this model to a ground plane, coloured based on infrared reflectivity from

LiDAR return intensities. They identify that their novel navigation system is up to an order of magnitude more accurate than a GNSS-aided IMU and odometry navigation system.

Our rationale for writing this paper is twofold. First, we believe the approach we propose for the integration of LiDAR measurement into a GNSS-aided IMU solution is novel. Second, we have repeatedly sought clear descriptions of the algorithms others have used, only to find missing details or gaps that have frustrated our implementation efforts. Here, we attempt to give a full and complete description of a loosely coupled GNSS-aided IMU navigation system in the expectation that it will be useful to others.

2. Experimental Platform

Figure 1 shows the experimental platform considered in this paper. The machine is called a track-type tractor or bulldozer and has applications in agriculture, construction, and mining. In all these domains, tracking spatial pose is needed for technology applications that include production reporting, guidance, and automation.

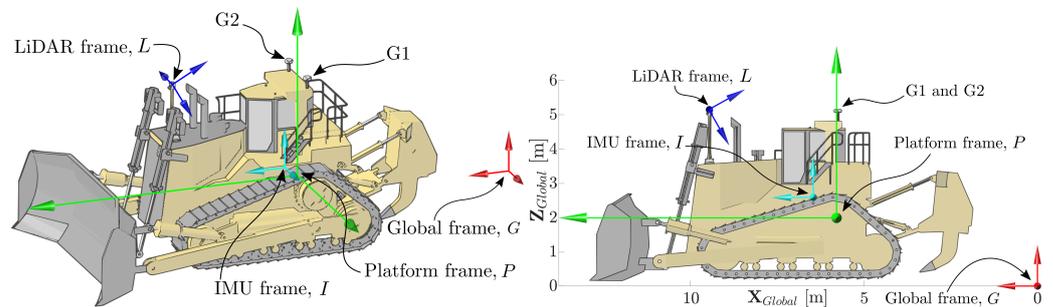


Figure 1. The Caterpillar D11T bulldozer considered in this paper, showing the sensors installed on the platform. The platform is equipped with two GNSS receivers, one IMU, and a scanning LiDAR sensor that faces forward and scans around a horizontal axis. The frames referred to in this paper are labelled as the LiDAR frame, L , the IMU frame, I , the platform frame, P , and the global frame, G .

Figure 1 shows the coordinate frames in which variables relevant to the development of the pose estimation system are represented. The global frame is denoted as G . Frame P describes the pose of the tractor in Frame G by a transformation $T_P(\cdot, \cdot)$. Sensors are described relative to Frame P : Frame I describes the position and orientation of the IMU; Frame L describes the position and orientation of a LiDAR; and G_1 and G_2 are the centres of the antennas of two GNSS receivers mounted on the tractor cabin.

The pose of the tractor is defined by the state vector,

$$\mathbf{x} = [\mathbf{p}_P \quad \dot{\mathbf{p}}_P \quad \ddot{\mathbf{p}}_P \quad \Phi_P \quad \omega_P \quad \mathbf{b}_a \quad \mathbf{b}_r]^T. \quad (1)$$

The 21 state parameters of the state vector are intended to accommodate a constant acceleration model for translation and a constant rate model for attitude. Here,

$\mathbf{p}_P = [x, y, z]^T$ is the coordinate of the origin of P in G ;

$\dot{\mathbf{p}}_P = [\dot{x}, \dot{y}, \dot{z}]^T$ is the velocity of point \mathbf{p}_P in G ;

$\ddot{\mathbf{p}}_P = [\ddot{x}, \ddot{y}, \ddot{z}]^T$ is the acceleration of point \mathbf{p}_P in G ;

$\Phi_P = [\theta, \phi, \psi]^T$ describes the attitude of Frame P in G as *roll*, *pitch* and *yaw*;

$\omega_P = [\omega_x, \omega_y, \omega_z]^T$ describes the rotational velocity of Frame P in G ;

$\mathbf{b}_a = [b_{\ddot{x}}, b_{\ddot{y}}, b_{\ddot{z}}]^T$ describes the acceleration bias of the IMU in Frame I ;

$\mathbf{b}_r = [b_{\omega_x}, b_{\omega_y}, b_{\omega_z}]^T$ is the rotational rate bias of the IMU in Frame I .

The transformation matrix $T_P(\Phi_P, \mathbf{p}_P)$ is given by

$$T_P(\Phi_P, \mathbf{p}_P) = \begin{bmatrix} R_P(\Phi_P) & \mathbf{p}_P \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}, \quad (2)$$

where

$$R_P(\Phi_P) = \begin{bmatrix} c(\psi)c(\phi) & c(\psi)s(\phi)s(\theta) - s(\psi)c(\theta) & c(\psi)s(\phi)c(\theta) + s(\psi)s(\theta) \\ s(\psi)c(\phi) & s(\psi)s(\phi)s(\theta) + c(\psi)c(\theta) & s(\psi)s(\phi)s(\theta) - c(\psi)s(\theta) \\ -s(\phi) & c(\phi)s(\theta) & c(\phi)c(\theta) \end{bmatrix}. \quad (3)$$

Here, $c(\cdot)$ is the cosine function and $s(\cdot)$ is the sine function. Note, $T_P(\cdot, \cdot)$ transforms points in Frame P into Frame G ; equally, it describes the location of Frame P in Frame G . The top left 3×3 block of the $T_P(\cdot, \cdot)$ matrix is the rotation matrix which aligns Frame P in Frame G .

The IMU provides measurements at 50 Hz; the GNSS receivers are RTK-enabled and provide observations at 10 Hz. The LiDAR is a Velodyne VLP-16 and provides range measurements from 16 beams, equally spaced across a field of view of 30° with 360° and rotation at 10 Hz. Other details of the sensors are given in Table 1.

Table 1. Specifications of the LiDAR, IMU, and GNSS sensors used as inputs to the navigation system of this paper.

Sensors					
Velodyne VLP-16 LiDAR Sensor [41]		IMU		GNSS Receivers	
		Accelerometers			
Number of rays:	16	Axes:	3	x uncertainty (1σ):	0.03 m
Range uncertainty (1σ):	0.03 m	Data Rate:	50 Hz	y uncertainty (1σ):	0.03 m
Horizontal field of view:	360°			z uncertainty (1σ):	0.055 m
Vertical field of view:	$30^\circ (\pm 15^\circ)$			Signals tracked:	L1, L2, L5
Horizontal resolution:	0.1° – 0.4°	Gyroscopes		Data rate:	10 Hz
Vertical resolution:	2°	Axes:	3	Corrections:	RTK
Scan rate:	5 Hz–20 Hz	Data Rate:	50 Hz		
Points per second:	300,000				

3. Pose Estimation Using GNSS, IMU, and LiDAR Measurements

The pose estimator has the structure given in Figure 2. Each block corresponds to a Kalman filter update. On receipt of a sensor measurement, a filter update occurs that depends on the type of the measurement (IMU, GNSS, LiDAR) with the structure designed to accommodate different data rates, including, prospectively, sequences of lost measurements. The structure takes advantage of the high-frequency IMU measurements, updated through an indirect Kalman filter, and the low-frequency GNSS and LiDAR measurements, which are updated through Kalman and Information filters, respectively. The structure depicted in Figure 2 is generally known as an indirect, or error-state, Kalman filter. It is the common method for implementing IMU-aided navigation solutions [42–50]. The formulation below assumes that IMU data arrive at a higher rate than either GNSS or LiDAR measurements.

Algorithm 1 gives the algorithm based on updates from the IMU, GNSS, and LiDAR measurements.

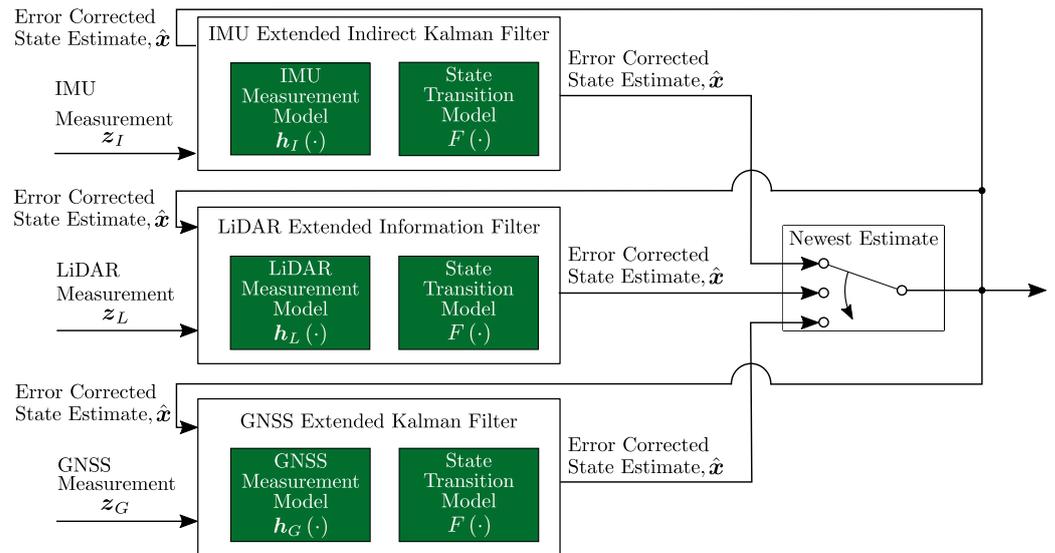


Figure 2. A diagrammatic representation of the proposed loosely coupled LiDAR- and GNSS-aided navigation system.

Algorithm 1: LiDAR- and GNSS aided IMU navigation solution algorithm.

Input: Initial state estimate, $\hat{x}_{0|0}$, and state estimate covariance, $\sigma_{\hat{x},0|0}^2$.

Initialise filter:

$k = 0$.

while Receiving measurements **do**

$k = k + 1$

$\Delta t_k = t_k - t_{k-1}$

switch Measurement type **do**

case IMU measurement **do**

$\left[\hat{x}_{k|k}, \sigma_{\hat{x},k|k}^2 \right] = \text{IMUUpdate}(\hat{x}_{k-1|k-1}, \sigma_{\hat{x},k-1|k-1}^2, z_{I,k}, \sigma_{z,I}^2, \Delta t_k)$.

case GNSS measurement **do**

$\left[\hat{x}_{k|k}, \sigma_{\hat{x},k|k}^2 \right] = \text{GNSSUpdate}(\hat{x}_{k-1|k-1}, \sigma_{\hat{x},k-1|k-1}^2, z_{G,k}, \sigma_{z,G}^2, \Delta t_k)$.

case LiDAR measurement **do**

$\left[\hat{x}_{k|k}, \sigma_{\hat{x},k|k}^2 \right] = \text{LiDARUpdate}(\hat{x}_{k-1|k-1}, \sigma_{\hat{x},k-1|k-1}^2, z_{L,k}, \sigma_{z,L}^2, \Delta t_k)$.

4. Process Model

The three filter updates of Figure 2 share the same process model which is assumed linear having the discrete-time form

$$\mathbf{x}_{k+1} = F(t_{k+1} - t_k) \cdot \mathbf{x}_k + \left[\int_{t_k}^{t_{k+1}} F(t_{k+1} - \tau) d\boldsymbol{\beta}(\tau) \right]. \quad (4)$$

The state transition model $F(\Delta t_{k+1}) = F(t_{k+1} - t_k)$ predicts the platform's state vector assuming a constant acceleration for translations and a constant rate for rotations. The state transition model also predicts the IMU acceleration and rotational rate biases through use of constants determined from experimental observations (see [51]). The 21×21 matrix $F(\cdot)$ has the form

$$F(\Delta t_k) = \begin{bmatrix} I_{3 \times 3} & \Delta t_k \cdot I_{3 \times 3} & \frac{\Delta t_k^2}{2} \cdot I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I_{3 \times 3} & \Delta t_k \cdot I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & I_{3 \times 3} & \Delta t_k \cdot I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & \frac{T_{b_{\bar{x},\bar{y},\bar{z}}}}{T_{b_{\bar{x},\bar{y},\bar{z}}} + \Delta t_k} \cdot I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & \frac{T_{b_{\omega_x, \omega_y, \omega_z}}}{T_{b_{\omega_x, \omega_y, \omega_z}} + \Delta t_k} \cdot I_{3 \times 3} \end{bmatrix}. \tag{5}$$

Here, $T_{b_{\bar{x},\bar{y},\bar{z}}}$ describes the empirically determined accelerometer bias parameters, and $T_{b_{\omega_x, \omega_y, \omega_z}}$ the empirically determined gyroscope bias parameters. Observe that

$$\lim_{\Delta t_k \rightarrow 0} F(\Delta t) = I_{21 \times 21}. \tag{6}$$

The second term of the right-hand side of Equation (4) describes the process noise which accounts for modelling approximations and model integration errors. We define

$$w(t_k) = \left[\int_{t_k}^{t_{k+1}} F(t_{k+1} - \tau) d\beta(\tau) \right], \tag{7}$$

where $w(\cdot)$ is a white Gaussian discrete-time process and $\beta(\cdot)$ is Brownian motion with diffusion $\sigma_Q^2(t)$ such that for consecutive measurement times t_k and t_{k+1} ,

$$E[\beta(t_k) - \beta(t_{k+1})] = 0, \tag{8}$$

$$E[\{\beta(t_k) - \beta(t_{k+1})\}\{\beta(t_k) - \beta(t_{k+1})\}^T] = \int_{t_k}^{t_{k+1}} \sigma_Q^2(t) dt. \tag{9}$$

If the sample period is assumed to be small compared to the auto-correlation times of the process model modes,

$$\sigma_{Q,k+1}^2 = E[w(t_k)w^T(t_k)] \approx \sigma_Q^2(t_k) \cdot [t_{k+1} - t_k]. \tag{10}$$

Observe that if the diffusion $\sigma_Q^2(t)$ is time-invariant, the process noise covariance, σ^2 , increases linearly from the time since the last measurement. As this time increases, the process noise increases, resulting in the process model being weighted less in the Kalman filter update and measurements being weighted more.

5. Pose State Updates from IMU Measurements

5.1. IMU Measurement Model

The IMU is installed on the platform frame close to the platform’s centre of gravity and is defined by Frame I —see Figure 3. IMU measurements are made in an inertial frame instantaneously aligned with Frame I . The IMU measurement model follows from Equation (3) and takes the form

$$z_I = h_I(x) + v_{z,I}, \tag{11}$$

where

$$z_I = [a_I \ \omega_I]^T = [\ddot{x}_I \ \ddot{y}_I \ \ddot{z}_I \ \dot{\theta}_I \ \dot{\phi}_I \ \dot{\psi}_I]^T \tag{12}$$

and measurement noise $v_{z,I}$ is assumed to be white and Gaussian with covariance $\sigma_{z,I}^2$ —see Table 2.

Table 2. The diagonal elements of the IMU measurement covariance matrix. The cross-terms are several orders of magnitude smaller than the diagonal element. These values were determined experimentally from statistical analysis of a stationary data segment of 930 s.

IMU Covariance, $\sigma_{z,I}^2$					
$\ddot{x}_I \left[\frac{\text{m}}{\text{s}^2} \right]^2$	$\ddot{y}_I \left[\frac{\text{m}}{\text{s}^2} \right]^2$	$\ddot{z}_I \left[\frac{\text{m}}{\text{s}^2} \right]^2$	$\dot{\theta}_I \left[\frac{\circ}{\text{s}} \right]^2$	$\dot{\phi}_I \left[\frac{\circ}{\text{s}} \right]^2$	$\dot{\psi}_I \left[\frac{\circ}{\text{s}} \right]^2$
7.8×10^{-5}	3.9×10^{-5}	4.0×10^{-4}	7.6×10^{-3}	7.6×10^{-3}	1.4×10^{-2}

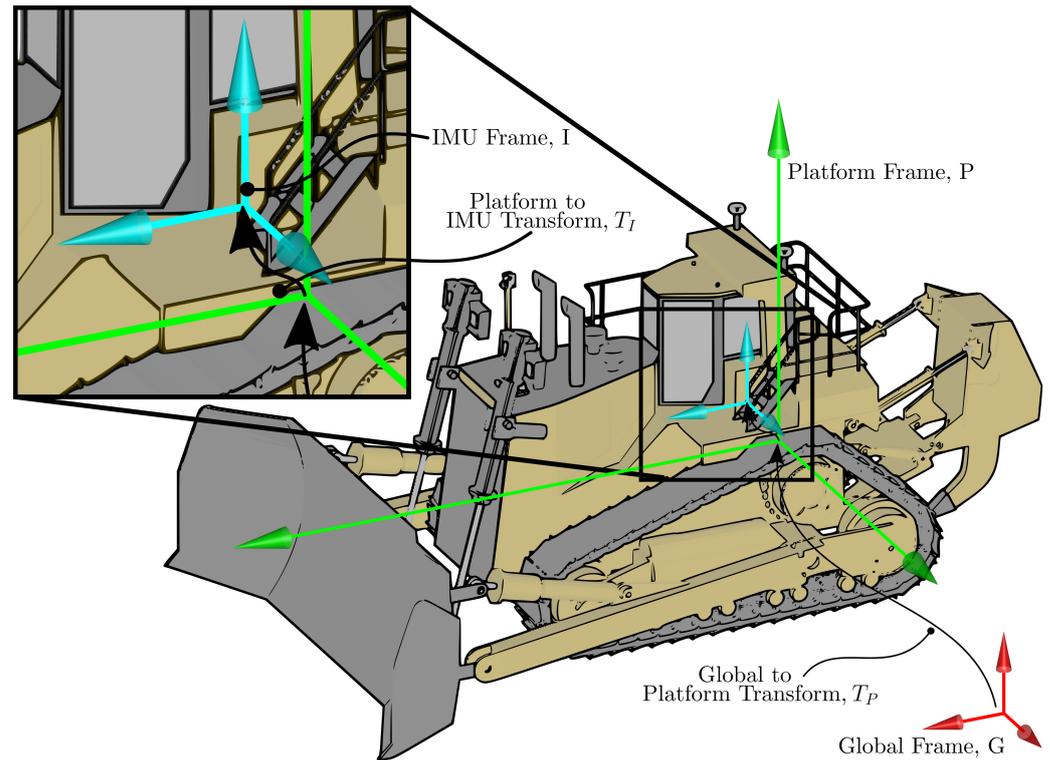


Figure 3. IMU Frame, I , relative to the platform Frame, P , and the global frame, G .

To determine the function $h_I(\cdot)$, note that the position of the origin of the IMU in the global frame is given by

$$\begin{bmatrix} \mathbf{p}_I(t) \\ 1 \end{bmatrix} = \begin{bmatrix} R_P(t) & \mathbf{p}_P(t) \\ 0_{1 \times 3} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{p}_i \\ 1 \end{bmatrix}, \quad (13)$$

where \mathbf{p}_i is the (fixed and known) position of the origin of Frame I in Frame P . The velocity of this point at time t is

$$\begin{bmatrix} \dot{\mathbf{p}}_I(t) \\ 0 \end{bmatrix} = \begin{bmatrix} \dot{R}_P(t) & \dot{\mathbf{p}}_P(t) \\ 0_{1 \times 3} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{p}_i \\ 1 \end{bmatrix} = \begin{bmatrix} \Omega_P(t) & \mathbf{v}_P(t) \\ 0_{1 \times 3} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{p}_i \\ 1 \end{bmatrix}, \quad (14)$$

where $\Omega_P(t)$ is the anti-symmetric matrix representing the angular velocity of Frame P in Frame G . The matrix $\Omega_P(t)$ is

$$\Omega_P = \text{sk}(\boldsymbol{\omega}_P) = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}. \quad (15)$$

The acceleration of the origin of Frame I in Frame G is found by differentiating Equation (14), i.e.,

$$\begin{bmatrix} \ddot{\mathbf{p}}_I \\ 0 \end{bmatrix} = \begin{bmatrix} \ddot{R}_P(t) & \dot{\mathbf{p}}_P(t) \\ 0_{1 \times 3} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{p}_i \\ 1 \end{bmatrix}. \quad (16)$$

Noting that if a constant angular rate is assumed (i.e., $\dot{\Omega}_P = 0$),

$$\ddot{R}_P(t) = \dot{\Omega}_P(t) \cdot R_P(t) + \Omega_P(t) \cdot \dot{R}_P(t) = \dot{\Omega}_P(t) \cdot R_P(t) + \Omega_P^2(t) \cdot R_P(t) = \Omega_P^2(t) \cdot R_P(t), \quad (17)$$

which gives,

$$\begin{bmatrix} \ddot{\mathbf{p}}_I(t) \\ 0 \end{bmatrix} = \begin{bmatrix} \Omega_P^2(t) \cdot R_P(t) & \dot{\mathbf{p}}_P(t) \\ 0_{1 \times 3} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{p}_i(t) \\ 1 \end{bmatrix}. \quad (18)$$

The measurement equations follow from Equation (18) and can be expressed in terms of the pose state vector:

$$\mathbf{z}_I = \mathbf{h}_I(\mathbf{x}) = \begin{bmatrix} \mathbf{a}_I \\ \boldsymbol{\omega}_I \end{bmatrix} = R_I^T \cdot R_P^T(\Phi_P) \cdot \begin{bmatrix} \Omega_P^2(\boldsymbol{\omega}_P) \cdot R_P(\Phi_P) \cdot \mathbf{p}_i + \dot{\mathbf{p}}_P + \mathbf{g} \\ \boldsymbol{\omega}_P \end{bmatrix} + \begin{bmatrix} \mathbf{b}_a \\ \mathbf{b}_r \end{bmatrix}. \quad (19)$$

Note that the rotation matrix $R_I^T \cdot R_P^T(\cdot)$ serves to align the acceleration and rate vectors to Frame I in which measurements are made. The vector \mathbf{b}_r is the bias of rate gyro, \mathbf{b}_a is the accelerometer bias, and \mathbf{g} denotes gravitational acceleration.

The Jacobian of $\mathbf{h}(\cdot)$ can be found analytically from

$$\nabla \mathbf{h}_I(\mathbf{x}) = \begin{bmatrix} 0_{3 \times 3} & 0_{3 \times 3} & I_{3 \times 3} & M_1 & M_2 & I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & R_I^T \cdot \frac{\partial R_P(\Phi_P)}{\partial \Phi_P} \cdot \boldsymbol{\omega}_P & R_I^T \cdot R_P^T(\Phi_P) & 0_{3 \times 3} & I_{3 \times 3} \end{bmatrix}, \quad (20)$$

where

$$\begin{aligned} M_1 &= R_I^T \cdot \frac{\partial R_P^T(\Phi_P)}{\partial \Phi_P} \cdot \left(\Omega_P^2(\boldsymbol{\omega}_P) \cdot R_P(\Phi_P) \cdot \mathbf{p}_i + \dot{\mathbf{p}}_P + \mathbf{g} \right) \\ &\quad + R_I^T \cdot R_P^T(\Phi_P) \cdot \Omega_P^2(\boldsymbol{\omega}_P) \cdot \frac{\partial R_P(\Phi_P)}{\partial \Phi_P} \cdot \mathbf{p}_i, \end{aligned} \quad (21)$$

$$M_2 = R_I^T \cdot R_P^T(\Phi_P) \cdot \frac{\partial \Omega_P^2(\boldsymbol{\omega}_P)}{\partial \boldsymbol{\omega}_P} \cdot R_P(\Phi_P) \cdot \mathbf{p}_i. \quad (22)$$

5.2. IMU Updates

The approach to updating measurements from the IMU involves estimating the error on the state estimate, $\delta \hat{\mathbf{x}}'_{k|k'}$ at time t_k given measurements to time $t_{k'}$ and combining this with the propagated state estimate, $\hat{\mathbf{x}}'_{k|k'}$ from the most recent GNSS or LiDAR update. The state estimate update, $\hat{\mathbf{x}}_{k|k}$, is then computed as the sum,

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}'_{k|k} + \delta \hat{\mathbf{x}}'_{k|k}. \quad (23)$$

The IMU filter is reset after each GNSS or LiDAR update: state estimate and state estimate covariance are set from the error-corrected state estimate and error-corrected state estimate covariance,

$$\hat{\mathbf{x}}'_{k-1|k-1} = \hat{\mathbf{x}}_{k-1|k-1}, \quad (24)$$

$$\boldsymbol{\sigma}_{\hat{\mathbf{x}}',k-1|k-1}^2 = \boldsymbol{\sigma}_{\hat{\mathbf{x}},k-1|k-1}^2 \quad (25)$$

and the error state estimate and error state estimate covariance are initialised to zero:

$$\delta \hat{\mathbf{x}}'_{k-1|k-1} = \mathbf{0}_{21 \times 1}, \quad (26)$$

$$\sigma_{\delta \hat{\mathbf{x}}',k-1|k-1}^2 = \mathbf{0}_{21 \times 21}. \quad (27)$$

On receipt of an IMU measurement, the calculation of $\delta \hat{\mathbf{x}}_{k|k}$ starts with the prediction of the state estimate,

$$\hat{\mathbf{x}}'_{k|k-1} = F(\Delta t_k) \cdot \hat{\mathbf{x}}'_{k-1|k-1}, \quad (28)$$

the error state,

$$\delta \hat{\mathbf{x}}'_{k|k-1} = F(\Delta t_k) \cdot \delta \hat{\mathbf{x}}'_{k-1|k-1}, \quad (29)$$

state covariance,

$$\sigma_{\hat{\mathbf{x}}',k|k-1}^2 = F(\Delta t_k) \cdot \sigma_{\hat{\mathbf{x}}',k-1|k-1}^2 \cdot F(\Delta t_k)^T. \quad (30)$$

and error state covariance,

$$\sigma_{\delta \hat{\mathbf{x}}',k|k-1}^2 = F(\Delta t_k) \cdot \sigma_{\delta \hat{\mathbf{x}}',k-1|k-1}^2 \cdot F(\Delta t_k)^T + \sigma_{\mathbf{Q},k}^2. \quad (31)$$

The innovation is computed from

$$\mathbf{v}_{I,k} = \mathbf{z}_{I,k} - \left(\mathbf{h}_I(\hat{\mathbf{x}}'_{k|k-1}) + \nabla \mathbf{h}_I(\hat{\mathbf{x}}'_{k|k-1}) \cdot \delta \hat{\mathbf{x}}'_{k|k-1} \right) = \mathbf{z}_{I,k} - \left(\hat{\mathbf{z}}_{I,\hat{\mathbf{x}}'_{k|k-1}} + \hat{\mathbf{z}}_{I,\delta \hat{\mathbf{x}}'_{k|k-1}} \right), \quad (32)$$

and the innovation covariance, $\sigma_{\mathbf{v},k}^2$, and Kalman gain, $W_{I,k}$, are found from

$$\sigma_{\mathbf{v},I,k}^2 = \nabla \mathbf{h}_I(\hat{\mathbf{x}}'_{k|k-1}) \cdot \sigma_{\delta \hat{\mathbf{x}}',k|k-1}^2 \cdot \nabla \mathbf{h}_I(\hat{\mathbf{x}}'_{k|k-1})^T + \sigma_{\mathbf{z},I}^2, \quad (33)$$

$$W_{I,k} = \sigma_{\delta \hat{\mathbf{x}}',k|k-1}^2 \cdot \nabla \mathbf{h}_I(\hat{\mathbf{x}}'_{k|k-1})^T \cdot \left(\sigma_{\mathbf{v},I,k}^2 \right)^{-1}. \quad (34)$$

The corrected error state and the corrected error state covariance are computed from

$$\delta \hat{\mathbf{x}}'_{k|k} = \delta \hat{\mathbf{x}}'_{k|k-1} + W_{I,k} \cdot \mathbf{v}_{I,k}, \quad (35)$$

$$\sigma_{\delta \hat{\mathbf{x}}',k|k}^2 = \sigma_{\delta \hat{\mathbf{x}}',k|k-1}^2 - W_{I,k} \cdot \sigma_{\mathbf{v},I,k}^2 \cdot W_{I,k}^T. \quad (36)$$

The predicted but uncorrected state estimate, $\hat{\mathbf{x}}_{k|k-1}$, and state estimate covariance, $\sigma_{\hat{\mathbf{x}},k|k-1}^2$, are used in the next iteration of the filter,

$$\hat{\mathbf{x}}'_{k|k} = \hat{\mathbf{x}}'_{k|k-1}, \quad (37)$$

$$\sigma_{\hat{\mathbf{x}}',k|k}^2 = \sigma_{\hat{\mathbf{x}}',k|k-1}^2. \quad (38)$$

The error-corrected state estimate, $\hat{\mathbf{x}}_{k|k}$, which incorporates the newest IMU observation, is computed through the addition with the error state estimate,

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}'_{k|k} + \delta \hat{\mathbf{x}}'_{k|k}, \quad (39)$$

$$\sigma_{\hat{\mathbf{x}},k|k}^2 = \sigma_{\hat{\mathbf{x}}',k|k}^2 + \sigma_{\delta \hat{\mathbf{x}}',k|k}^2. \quad (40)$$

The indirect extended Kalman filter (IEKF) algorithm for IMU updates is detailed in Algorithm 2.

Algorithm 2: IMUUpdate - An Indirect Extended Kalman Filter (IEKF) algorithm for state estimation using IMU observations.

Function $[\hat{\mathbf{x}}_{k|k}, \sigma_{\hat{\mathbf{x}},k|k}^2] = \text{IMUUpdate}(\hat{\mathbf{x}}_{k-1|k-1}, \sigma_{\hat{\mathbf{x}},k-1|k-1}^2, \mathbf{z}_{I,k}, \sigma_{\mathbf{z},I}^2, \Delta t_k)$:

if the previous measurement was from the GNSS or LiDAR **then**

 Initialise the filter with the following static variables:

$$\begin{aligned}\hat{\mathbf{x}}'_{k-1|k-1} &= \hat{\mathbf{x}}_{k-1|k-1}, \\ \sigma_{\hat{\mathbf{x}}',k-1|k-1}^2 &= \sigma_{\hat{\mathbf{x}},k-1|k-1}^2, \\ \delta\hat{\mathbf{x}}'_{k-1|k-1} &= \mathbf{0}_{21 \times 1}, \\ \sigma_{\delta\hat{\mathbf{x}}',k-1|k-1}^2 &= \mathbf{0}_{21 \times 21}.\end{aligned}$$

Predict the state estimate: $\hat{\mathbf{x}}'_{k|k-1} = F(\Delta t_k) \cdot \hat{\mathbf{x}}'_{k-1|k-1}$.

Predict the error state estimate: $\delta\hat{\mathbf{x}}'_{k|k-1} = F(\Delta t_k) \cdot \delta\hat{\mathbf{x}}'_{k-1|k-1}$.

Predict the state estimate covariance: $\sigma_{\hat{\mathbf{x}}',k|k-1}^2 = F(\Delta t_k) \cdot \sigma_{\hat{\mathbf{x}}',k-1|k-1}^2 \cdot F(\Delta t_k)^T$.

Predict the error state covariance:

$$\sigma_{\delta\hat{\mathbf{x}}',k|k-1}^2 = F(\Delta t_k) \cdot \sigma_{\delta\hat{\mathbf{x}}',k-1|k-1}^2 \cdot F(\Delta t_k)^T + \sigma_{\mathbf{Q},k}^2(\Delta t_k).$$

Compute the innovation: $\mathbf{v}_{I,k} = \mathbf{z}_{I,k} - (\mathbf{h}_I(\hat{\mathbf{x}}'_{k|k-1}) + \nabla \mathbf{h}_I(\hat{\mathbf{x}}'_{k|k-1}) \cdot \delta\hat{\mathbf{x}}'_{k|k-1})$.

Compute innovation covariance:

$$\sigma_{\mathbf{v},I,k}^2 = \nabla \mathbf{h}_I(\hat{\mathbf{x}}'_{k|k-1}) \cdot \sigma_{\delta\hat{\mathbf{x}}',k|k-1}^2 \cdot \nabla \mathbf{h}_I(\hat{\mathbf{x}}'_{k|k-1})^T + \sigma_{\mathbf{z},I}^2.$$

Compute the Kalman gain: $W_{I,k} = \sigma_{\delta\hat{\mathbf{x}}',k|k-1}^2 \cdot \nabla \mathbf{h}_I(\hat{\mathbf{x}}'_{k|k-1})^T \cdot (\sigma_{\mathbf{v},I,k}^2)^{-1}$.

Update the error state estimate: $\delta\hat{\mathbf{x}}'_{k|k} = \delta\hat{\mathbf{x}}'_{k|k-1} + W_{I,k} \cdot \mathbf{v}_{I,k}$.

Update the error state estimate covariance:

$$\sigma_{\delta\hat{\mathbf{x}}',k|k}^2 = \sigma_{\delta\hat{\mathbf{x}}',k|k-1}^2 - W_{I,k} \cdot \sigma_{\mathbf{v},I,k}^2 \cdot W_{I,k}^T.$$

Update the state estimate covariance: $\sigma_{\hat{\mathbf{x}}',k|k}^2 = \sigma_{\hat{\mathbf{x}}',k|k-1}^2 + \sigma_{\delta\hat{\mathbf{x}}',k|k}^2$.

The predicted but uncorrected state estimate and state estimate covariance are used in the next iteration of the filter:

$$\begin{aligned}\hat{\mathbf{x}}'_{k|k} &= \hat{\mathbf{x}}'_{k|k-1}, \\ \sigma_{\hat{\mathbf{x}}',k|k}^2 &= \sigma_{\hat{\mathbf{x}}',k|k-1}^2.\end{aligned}$$

Compute the error-corrected state estimate: $\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}'_{k|k} + \delta\hat{\mathbf{x}}'_{k|k}$.

Compute the error-corrected state covariance: $\sigma_{\hat{\mathbf{x}},k|k}^2 = \sigma_{\hat{\mathbf{x}}',k|k}^2 + \sigma_{\delta\hat{\mathbf{x}}',k|k}^2$.

return $\hat{\mathbf{x}}_{k|k}$ and $\sigma_{\hat{\mathbf{x}},k|k}^2$.

6. Pose State Updates from GNSS Measurements

6.1. GNSS Measurement Model

The GNSS measurement model gives the locations of the GNSS receivers in the global frame and takes the form

$$\mathbf{z}_G = \mathbf{h}_G(\mathbf{x}) + \mathbf{v}_{z,G}, \quad (41)$$

Here, $\mathbf{h}_G(\cdot)$ gives the locations of the GNSS receivers as a function of the state,

$$\mathbf{z}_G = \mathbf{h}_G(\mathbf{x}) = \begin{bmatrix} R_P(\Phi_P) \cdot \mathbf{p}_{G1} + \mathbf{p}_P \\ R_P(\Phi_P) \cdot \mathbf{p}_{G2} + \mathbf{p}_P \end{bmatrix}, \quad (42)$$

where \mathbf{p}_{G1} and \mathbf{p}_{G2} are the coordinates of the GNSS antennas in the platform frame (see Figure 4). The measurement vector \mathbf{z}_G is the stacked vectors of the antennas' Cartesian coordinates in Frame G. Equation (42) can be readily extended if a third GNSS receiver

were mounted to the platform. The measurement noise, $v_{z,G}$, is assumed to be white and Gaussian with covariance $\sigma_{z,G}^2$ (see Table 3).

Table 3. The diagonal elements of the GNSS measurement covariance matrix. The off-diagonal terms are several orders of magnitude smaller than the diagonal elements. These values were determined from statistical analysis of a 930 s measurement sequence where the platform was stationary.

GNSS Covariance, $\sigma_{z,G}^2$		
$x_G[m]^2$	$y_G[m]^2$	$z_G[m]^2$
1.9×10^{-6}	2.4×10^{-6}	2.2×10^{-5}

The GNSS measurement Jacobian is given by

$$\nabla h_G(x) = \begin{bmatrix} I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & \frac{\partial R_P^T(\Phi_P)}{\partial \Phi_P} \cdot p_{G1} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & \frac{\partial R_P^T(\Phi_P)}{\partial \Phi_P} \cdot p_{G2} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix}. \quad (43)$$

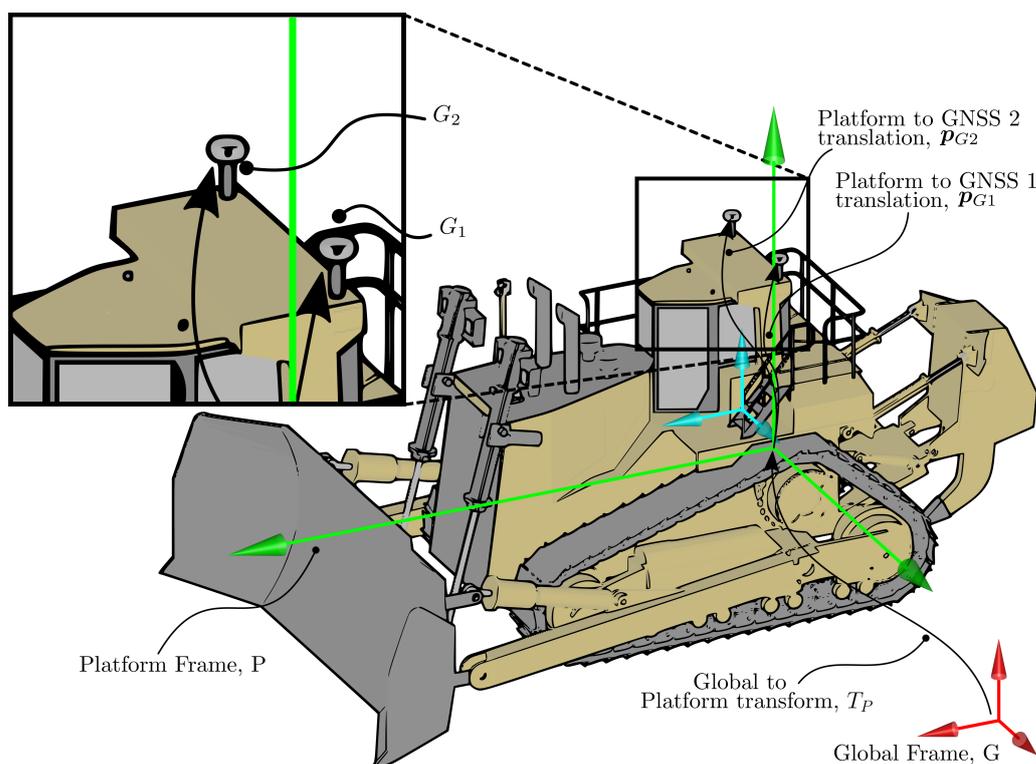


Figure 4. The locations of GNSS 1, p_{G2} , from the platform frame, P .

6.2. GNSS Updates

The estimation of the error-corrected state estimate, $\hat{x}_{k|k}$, begins with a prediction of state using a state transition model, $F(\Delta t_k)$, and the previous error-corrected state estimate, $\hat{x}_{k-1|k-1}$,

$$\hat{x}_{k|k-1} = F(\Delta t_k) \cdot (\hat{x}_{k-1|k-1}), \quad (44)$$

and the prediction of the state covariance,

$$\sigma_{\hat{x},k|k-1}^2 = F(\Delta t_k) \cdot \sigma_{\hat{x},k-1|k-1}^2 \cdot F(\Delta t_k)^T + \sigma_{Q,k}^2. \quad (45)$$

The GNSS measurement innovation and innovation covariance are computed from

$$\mathbf{v}_{G,k} = \mathbf{z}_{G,k} - \mathbf{h}_G(\hat{\mathbf{x}}_{k|k-1}) = \mathbf{z}_{G,k} - \hat{\mathbf{z}}_{G,k}, \quad (46)$$

$$\sigma_{\mathbf{v},G,k}^2 = \nabla \mathbf{h}_G(\hat{\mathbf{x}}_{k|k-1}) \cdot \sigma_{\hat{\mathbf{x}},k|k-1}^2 \cdot \nabla \mathbf{h}_G(\hat{\mathbf{x}}_{k|k-1})^T + \sigma_{\mathbf{z},G}^2, \quad (47)$$

where $\mathbf{z}_{G,k}$ is the GNSS measurement.

The Kalman gain, $W_{G,k}$, is then computed using

$$W_{G,k} = \sigma_{\hat{\mathbf{x}},k|k-1}^2 \cdot \nabla \mathbf{h}_G(\hat{\mathbf{x}}_{k|k-1})^T \cdot (\sigma_{\mathbf{v},G,k}^2)^{-1}. \quad (48)$$

The error-corrected state estimate, $\hat{\mathbf{x}}_{k|k}$, and error-corrected state estimate covariance, $\sigma_{\hat{\mathbf{x}},k|k}^2$ are computed using

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + W_{G,k} \cdot \mathbf{v}_{G,k}, \quad (49)$$

$$\sigma_{\hat{\mathbf{x}},k|k}^2 = \sigma_{\hat{\mathbf{x}},k|k-1}^2 - W_{G,k} \cdot \sigma_{\mathbf{v},G,k}^2 \cdot W_{G,k}^T. \quad (50)$$

The GNSS update is given as Algorithm 3.

Algorithm 3: GNSSUpdate—An extended Kalman filter (EKF) algorithm for state estimation using GNSS observations.

Function $[\hat{\mathbf{x}}_{k|k}, \sigma_{\hat{\mathbf{x}},k|k}^2] = \text{GNSSUpdate}(\hat{\mathbf{x}}_{k-1|k-1}, \sigma_{\hat{\mathbf{x}},k-1|k-1}^2, \mathbf{z}_{G,k}, \sigma_{\mathbf{z},G}^2, \Delta t_k)$:

Predict the state estimate: $\hat{\mathbf{x}}_{k|k-1} = F(\Delta t_k) \cdot \hat{\mathbf{x}}_{k-1|k-1}$.

Predict the state estimate covariance:

$$\sigma_{\hat{\mathbf{x}},k|k-1}^2 = F(\Delta t_k) \cdot \sigma_{\hat{\mathbf{x}},k-1|k-1}^2 \cdot F(\Delta t_k)^T + \sigma_{Q,k}^2(\Delta t_k).$$

Compute the innovation: $\mathbf{v}_{G,k} = \mathbf{z}_{G,k} - \mathbf{h}_G(\hat{\mathbf{x}}_{k|k-1}) = \mathbf{z}_{G,k} - \hat{\mathbf{z}}_{G,k}$.

Compute innovation covariance:

$$\sigma_{\mathbf{v},G,k}^2 = \nabla \mathbf{h}_G(\hat{\mathbf{x}}_{k|k-1}) \cdot \sigma_{\hat{\mathbf{x}},k|k-1}^2 \cdot \nabla \mathbf{h}_G(\hat{\mathbf{x}}_{k|k-1})^T + \sigma_{\mathbf{z},G}^2.$$

Compute the Kalman gain: $W_{G,k} = \sigma_{\hat{\mathbf{x}},k|k-1}^2 \cdot \nabla \mathbf{h}_G(\hat{\mathbf{x}}_{k|k-1})^T \cdot (\sigma_{\mathbf{v},G,k}^2)^{-1}$.

Update the error-corrected state estimate: $\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + W_{G,k} \cdot \mathbf{v}_{G,k}$.

Update the error-corrected state estimate covariance:

$$\sigma_{\hat{\mathbf{x}},k|k}^2 = \sigma_{\hat{\mathbf{x}},k|k-1}^2 - W_{G,k} \cdot \sigma_{\mathbf{v},G,k}^2 \cdot W_{G,k}^T.$$

return $\hat{\mathbf{x}}_{k|k}$ and $\sigma_{\hat{\mathbf{x}},k|k}^2$.

7. Pose State Updates Using LiDAR Measurements

7.1. LiDAR Measurement Model in a Known Environment

The LiDAR measurement model developed in this section computes the expected range measurements based on the state vector and a triangulated model of the environment, denoted X . The LiDAR measurement model ray-casts from the LiDAR to the environment model X to determine the expected measurement ranges from the LiDAR sensor. In this section, the terrain model X is assumed known; in Section 9, the approach is generalised to unknown terrain.

The LiDAR measurement function is denoted

$$\mathbf{z}_L = \mathbf{h}_L(\mathbf{x}, X) + \mathbf{v}_{z,L}, \quad (51)$$

where the LiDAR measurement vector, \mathbf{z}_L is a set of ranges corresponding to each ray from the LiDAR sensor. For n ranges,

$$\mathbf{z}_L = [z_0 \quad z_1 \quad \cdots \quad z_n]^T. \quad (52)$$

The measurement noise $v_{z,L}$ is assumed to be white and Gaussian with covariance $\sigma_{z,L}^2$ —see Section 7.2.

The LiDAR measurement function, $h_L(\cdot, \cdot)$, determines the expected sensor observations by ray-casting against a known terrain model given a pose state estimate, X (see Figure 5). Each ray emanates from a point s_0 in Frame L known as the sensor origin. The direction of each ray or beam of the LiDAR can be defined by a second point s_j , $j = 1 \dots n$ with $\|s_j - s_0\| = 1$. The location of these points in Frame G can be found with transformations

$$q_0 = T_P(\Phi_P, p_P) \cdot T_L \cdot s_0, \quad (53)$$

$$q_j = T_P(\Phi_P, p_P) \cdot T_L \cdot s_j. \quad (54)$$

Here, T_L describes the location of Frame L in Frame P . This transform is assumed known. See [52] for approaches to finding it.

Each ray is parametrised by a range r_j and can be described by

$$\Gamma(r_j) = q_0 + r_j \cdot (q_j - q_0). \quad (55)$$

To find r_j , it is necessary to first determine the triangle in X which it intersects. Denote the vertices of this triangle, $\Delta_j = (\alpha_j, \beta_j, \gamma_j)$. Then,

$$r_j = \frac{\mathbf{n}_j \cdot (\alpha_j - q_0)}{\mathbf{n}_j \cdot (q_0 - q_j)}, \quad (56)$$

where

$$\mathbf{n}_j = (\beta_j - \alpha_j) \times (\gamma_j - \alpha_j). \quad (57)$$

The ray-triangle intersection is depicted in Figure 6.

The measurement model for n ray intersections can be constructed from

$$z_L = [r_1, r_2, \dots, r_n]^T, \quad (58)$$

and the LiDAR measurement Jacobian from

$$\nabla h_L(x, X) = \begin{bmatrix} \frac{\partial r_1}{\partial p_P} & 0_{1 \times 6} & \frac{\partial r_1}{\partial \Phi_P} & 0_{1 \times 9} \\ \frac{\partial r_2}{\partial p_P} & 0_{1 \times 6} & \frac{\partial r_2}{\partial \Phi_P} & 0_{1 \times 9} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial r_n}{\partial p_P} & 0_{1 \times 6} & \frac{\partial r_n}{\partial \Phi_P} & 0_{1 \times 9} \end{bmatrix}. \quad (59)$$

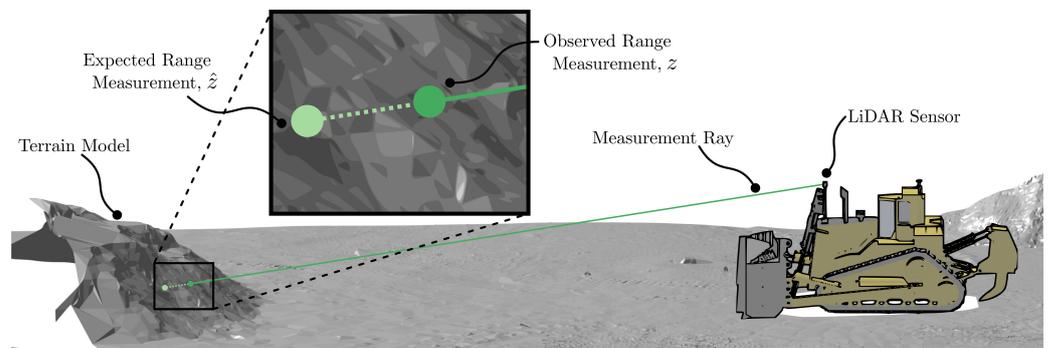


Figure 5. The LiDAR measurement model provides the measurement that the LiDAR sensor is expected to observe given the current state vector. The difference between the expected and observed measurements is termed the innovation.

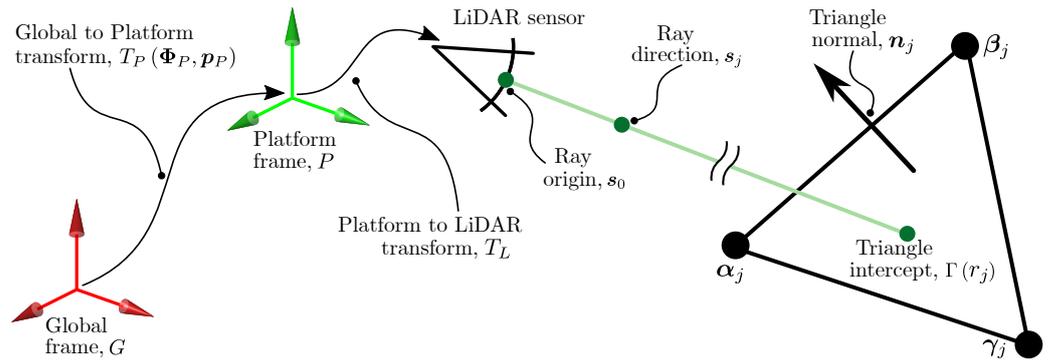


Figure 6. The measurement model for the LiDAR calculates the expected range measurement as the distance from the ray origin to the intercepted triangle of the model.

7.2. LiDAR Measurement Noise Model

Observations from the LiDAR sensor present in this application are transformed from the LiDAR frame, L , into the global frame, G . However, each of the transforms are required to achieve this contain uncertainty. For sensors such as the GNSS receivers and IMU, the uncertainty introduced by these transforms is small, as observations are obtained at the sensor. However, for perception sensors such as LiDAR, which perceive the environment at potentially significant ranges, small transform errors can lead to quite significant observation errors. For example, a registration error of 1° at 50 m results in ~ 0.7 m of endpoint error.

The uncertainty of frame locations and measurements may be propagated from their origin to the point of measurement to determine the effective uncertainty. We consider four sources of uncertainty which we propagate from their origin to the LiDAR ray endpoint (shown in Figure 7). The sources of uncertainty considered are (i) the uncertainty in the global frame location, σ_G^2 ; (ii) the estimated location of the platform relative to the global frame, σ_P^2 ; (iii) the uncertainty in the LiDAR registration, σ_L^2 ; and (iv) the LiDAR's measurement uncertainty, $\sigma_{z,L}^2$. The Jacobian of each of the relevant transforms is used to transform the uncertainty through the system.

The uncertainty from each of these sources propagates through to the endpoint of the ray as follows:

$$\sigma_{P,k}^2 = \nabla T_P(\Phi_P, \mathbf{p}_P) \cdot \sigma_G^2 \cdot \nabla T_P(\Phi_P, \mathbf{p}_P)^T + \sigma_P^2, \quad (60)$$

$$\sigma_{L,k}^2 = \nabla T_L \cdot \sigma_{P,k}^2 \cdot \nabla T_L^T + \sigma_L^2, \quad (61)$$

$$\sigma_{z,L}^2 = \nabla T_R \cdot \sigma_{L,k}^2 \cdot \nabla T_R^T + \sigma_{z,L}^2. \quad (62)$$

Here, the notation ∇T_L is used to denote the Jacobian of the homogeneous transform—in this case, the transform from the platform frame to the LiDAR frame.

Figure 7 provides a visual representation of the propagation of uncertainty from the global frame through to the endpoint of the ray. The covariances shown in Figure 7 have been exaggerated for visual clarity.

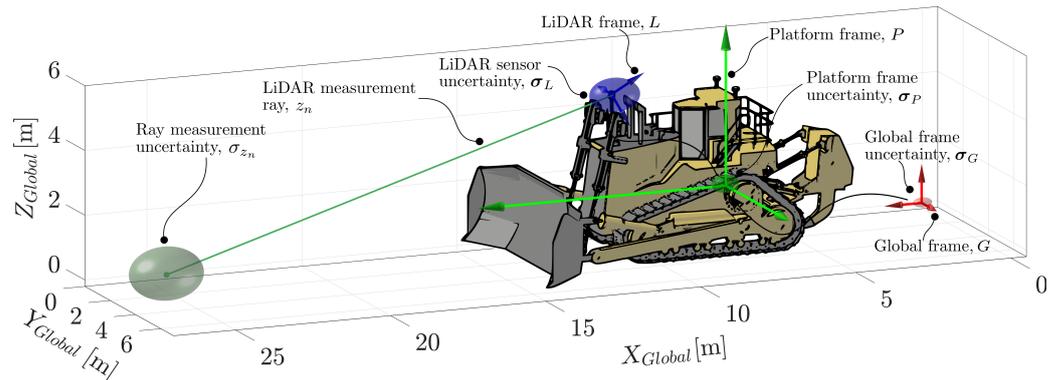


Figure 7. The location of each of the frames of the platform are uncertain to varying degrees. This uncertainty is propagated through the transforms of the system, originating at the global frame and ending at the ray endpoint. The propagated uncertainty is the effective uncertainty of that ray's endpoint.

7.3. LiDAR Updates

The number of LiDAR measurements typically exceeds the dimension of the state vector so it is computationally more efficient to use the information form of the extended Kalman filter to perform the pose update from LiDAR measurements [53].

The current state and covariance are transformed in the information space

$$Y_{k-1|k-1} = \left(\sigma_{\hat{x}_{k-1|k-1}}^2 \right)^{-1}, \quad (63)$$

$$\hat{y}_{k-1|k-1} = Y_{k-1|k-1} \cdot \hat{x}_{k-1|k-1}. \quad (64)$$

The Fisher information matrix (FIM) and Fisher information vector (FIV) are predicted to the current time step using

$$Y_{k|k-1} = \left[F(\Delta t_k) \cdot \left(Y_{k-1|k-1} \right)^{-1} \cdot F(\Delta t_k)^T + \sigma_{Q,k}^2 \right]^{-1}, \quad (65)$$

$$\hat{y}_{k|k-1} = Y_{k|k-1} \cdot F(\Delta t_k) \cdot \hat{x}_{k-1|k-1}. \quad (66)$$

The predicted state estimate is obtained from

$$\hat{x}_{k|k-1} = \left(Y_{k-1|k-1} \right)^{-1} \cdot \hat{y}_{k|k-1} \quad (67)$$

and the innovation is obtained using the LiDAR observation function, the current predicted state estimate, and the terrain model, X :

$$v_{L,k} = z_{L,k} - h_L \left(\hat{x}_{k|k-1}, X \right). \quad (68)$$

The predicted FIV and FIM are corrected through use of the innovation and the LiDAR range measurements using

$$\hat{y}_{k|k} = \hat{y}_{k|k-1} + \nabla h_L \left(\hat{x}_{k|k-1}, X \right)^T \cdot \left(\sigma_{z,L}^2 \right)^{-1} \cdot \left[v_{L,k} + \nabla h_L \left(\hat{x}_{k|k-1}, X \right) \cdot \hat{x}_{k|k-1} \right], \quad (69)$$

$$Y_{k|k} = Y_{k|k-1} + \nabla h_L \left(\hat{x}_{k|k-1}, X \right)^T \cdot \left(\sigma_{z,L}^2 \right)^{-1} \cdot \nabla h_L \left(\hat{x}_{k|k-1}, X \right). \quad (70)$$

The FIM and FIV are now transformed back into the state space,

$$\hat{\mathbf{x}}_{k|k} = \left(Y_{k|k} \right)^{-1} \cdot \hat{\mathbf{y}}_{k|k}, \quad (71)$$

$$\sigma_{\hat{\mathbf{x}},k|k}^2 = \left(Y_{k|k} \right)^{-1}. \quad (72)$$

Algorithm 4 shows the LiDAR pose estimation extended information filter (EIF) algorithmically.

Algorithm 4: LiDARUpdate—An extended information filter (EIF) algorithm for state estimation using LiDAR observations in a known environment.

Function $\left[\hat{\mathbf{x}}_{k|k}, \sigma_{\hat{\mathbf{x}},k|k}^2 \right] = \text{LiDARUpdate}(\hat{\mathbf{x}}_{k-1|k-1}, \sigma_{\hat{\mathbf{x}},k-1|k-1}^2, z_{L,k}, \sigma_{z,L}^2, \Delta t_k)$:

Transform the state estimate and state estimate covariance into the Information space:

$$Y_{k-1|k-1} = \left(\sigma_{\hat{\mathbf{x}},k-1|k-1}^2 \right)^{-1},$$

$$\hat{\mathbf{y}}_{k-1|k-1} = Y_{k-1|k-1} \cdot \hat{\mathbf{x}}_{k-1|k-1}.$$

Predict the Fisher information matrix:

$$Y_{k|k-1} = \left[F(\Delta t_k) \cdot \left(Y_{k-1|k-1} \right)^{-1} \cdot F(\Delta t_k)^T + \sigma_{Q,k}^2(\Delta t_k) \right]^{-1}.$$

Predict the Fisher information vector: $\hat{\mathbf{y}}_{k|k-1} = Y_{k|k-1} \cdot F(\Delta t_k) \cdot \hat{\mathbf{x}}_{k-1|k-1}$.

Compute the predicted state estimate: $\hat{\mathbf{x}}_{k|k-1} = \left(Y_{k-1|k-1} \right)^{-1} \cdot \hat{\mathbf{y}}_{k|k-1}$.

if in an unknown environment then

 Extract the terrain state elements, \mathbf{x}_t , from the state vector and triangulate them: $X = \text{triangulateGrid}(\mathbf{x}_t)$

 Compute the innovation: $\mathbf{v}_{L,k} = z_{L,k} - \mathbf{h}_L(\hat{\mathbf{x}}_{k|k-1}, X)$.

 Compute the uncertainty of the LiDAR range measurements using Section 7.2.

 Correct the predictions:

$$\hat{\mathbf{y}}_{k|k} = \hat{\mathbf{y}}_{k|k-1} + \nabla \mathbf{h}_L(\hat{\mathbf{x}}_{k|k-1}, X)^T \cdot \left(\sigma_{z,L}^2 \right)^{-1} \cdot \left[\mathbf{v}_{L,k} + \nabla \mathbf{h}_L(\hat{\mathbf{x}}_{k|k-1}, X) \cdot \hat{\mathbf{x}}_{k|k-1} \right],$$

$$Y_{k|k} = Y_{k|k-1} + \nabla \mathbf{h}_L(\hat{\mathbf{x}}_{k|k-1}, X)^T \cdot \left(\sigma_{z,L}^2 \right)^{-1} \cdot \nabla \mathbf{h}_L(\hat{\mathbf{x}}_{k|k-1}, X).$$

Transform the updated Fisher Information Vector and Fisher Information Matrix back into the state space:

$$\hat{\mathbf{x}}_{k|k} = \left(Y_{k|k} \right)^{-1} \cdot \hat{\mathbf{y}}_{k|k},$$

$$\sigma_{\hat{\mathbf{x}},k|k}^2 = \left(Y_{k|k} \right)^{-1}.$$

return $\hat{\mathbf{x}}_{k|k}$ and $\sigma_{\hat{\mathbf{x}},k|k}^2$.

8. Evaluation

It is necessary to use indirect methods for the evaluation of the pose estimator as there is no ground truth pose trajectory. The approach chosen is to survey the environment to obtain a ground truth model and compare LiDAR measurements with the predicted ranges from ray-casting against this scanned ground truth model. These differences are representative of the error in the pose. The survey to obtain the ground truth model was created with a Faro Focus^{3D} terrestrial LiDAR scanner [54]. This instrument claims a one standard deviation range precision of 0.0011 m. Figure 8 shows the environment model obtained by scanning.

The scene used for evaluation is shown in Figure 9. It comprises a generally flat terrain with piled dirt, forming mounds. The bulldozer was manually moved around in this environment following the path shown in Figure 8. During the execution of this motion, at various points, the platform was pitched and rolled using the bulldozer blade.

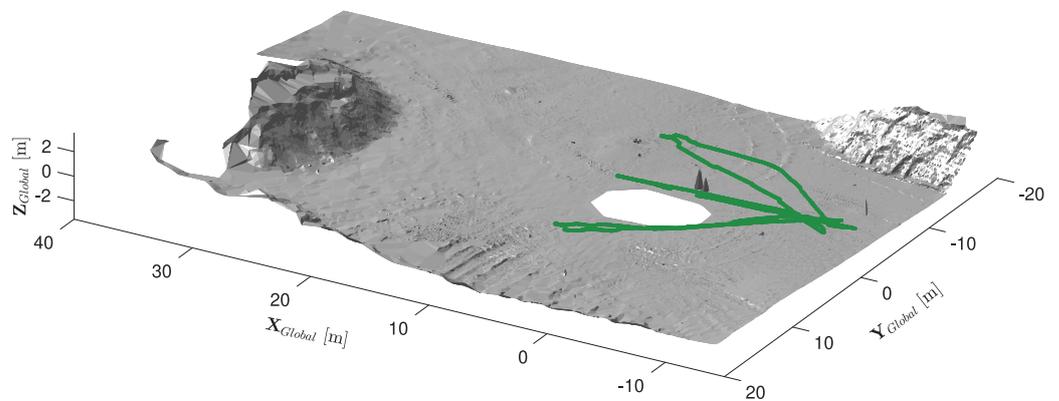


Figure 8. The environment in which the bulldozer platform manoeuvred is shown. The trajectory of the platform is shown as a green line. The platform exercised its rotational degrees of freedom through the use of the blade during this trajectory.



Figure 9. A photo of the environment in which the platform traversed is shown. Here, the bulldozer platform is visible in the centre of the image. The mound visible to the left of Figure 8 is visible to the right of this image.

Figure 10 shows the point-to-model errors for the GNSS-aided IMU solution and the LiDAR- and GNSS-aided solution in a known environment. A ray having the length of the measurement is cast along the corresponding LiDAR beam at the estimated pose. The distance from the end point of the ray to the nearest point of the model is determined as the point-to-model error. The smaller the point-to-model error, the closer the real pose to the estimated pose. For perfect pose, a perfect model, and zero measurement error, the point-to-model error is zero.

The distribution of point-to-model errors given in Figure 10 evidences that the LiDAR- and GNSS-aided IMU solution is more accurate than the GNSS aiding. The mean point-to-model error is reduced from 0.25 m to 0.06 m. Significantly, the highest number of point-to-model errors for the LiDAR- and GNSS-aided solution corresponds to zero error. This is not unexpected: the positioning of the GNSS antennas provides information that supports good spatial positioning along with roll and yaw estimation. However, no information is available from GNSS measurements about the orientation of the axis that passes through the two antenna centres. This rotation heavily contributes to the pitch of the platform. The provision of measurements of the forward-facing 3D LiDAR provides information that stabilises the pitch estimate, this accounting for the improved accuracy. The mean value for point-to-model error for the GNSS-aided solution is prospectively due to start-up bias on the centre. It is noted that the GNSS-aided solution has a broader distribution of error.

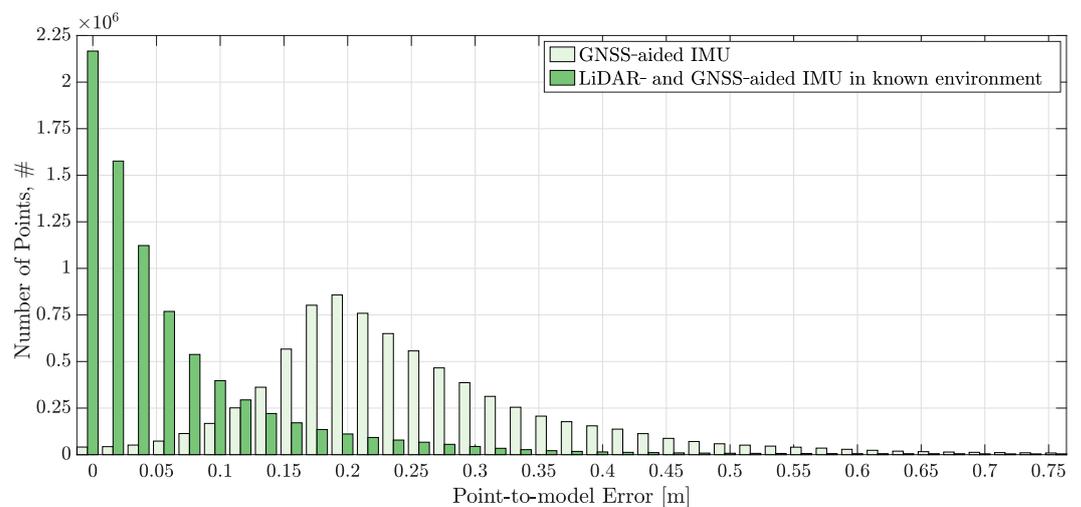


Figure 10. The point-to-model error distributions obtained by comparing the LiDAR endpoints with the ground truth terrain model using the GNSS-aided IMU and LiDAR- and GNSS-aided IMU navigation solutions.

The time estimates for the six degrees of freedom of the platform's pose are shown in Figure 11. Significantly both estimators give similar results for x , y , z , and yaw. Most of the difference is in estimates for the pitch and roll of the platform. The bias in the pitch estimate for the GNSS-aided solution is attributed to start-up bias in the IMU sensor, which the estimator has been unable to account for; it is present at both the start and end of the motion sequence when the platform is stationary.

The results in this paper were produced on a computer running a Linux-based operating system (Intel Xeon W-2125 CPU, 32 GB of memory). The GNSS-aided IMU filter could update up to $\sim 10,000$ Hz, much faster than the measurement rate. The LiDAR- and GNSS-aided IMU filter required more computing per time step and was capable of ~ 500 Hz.

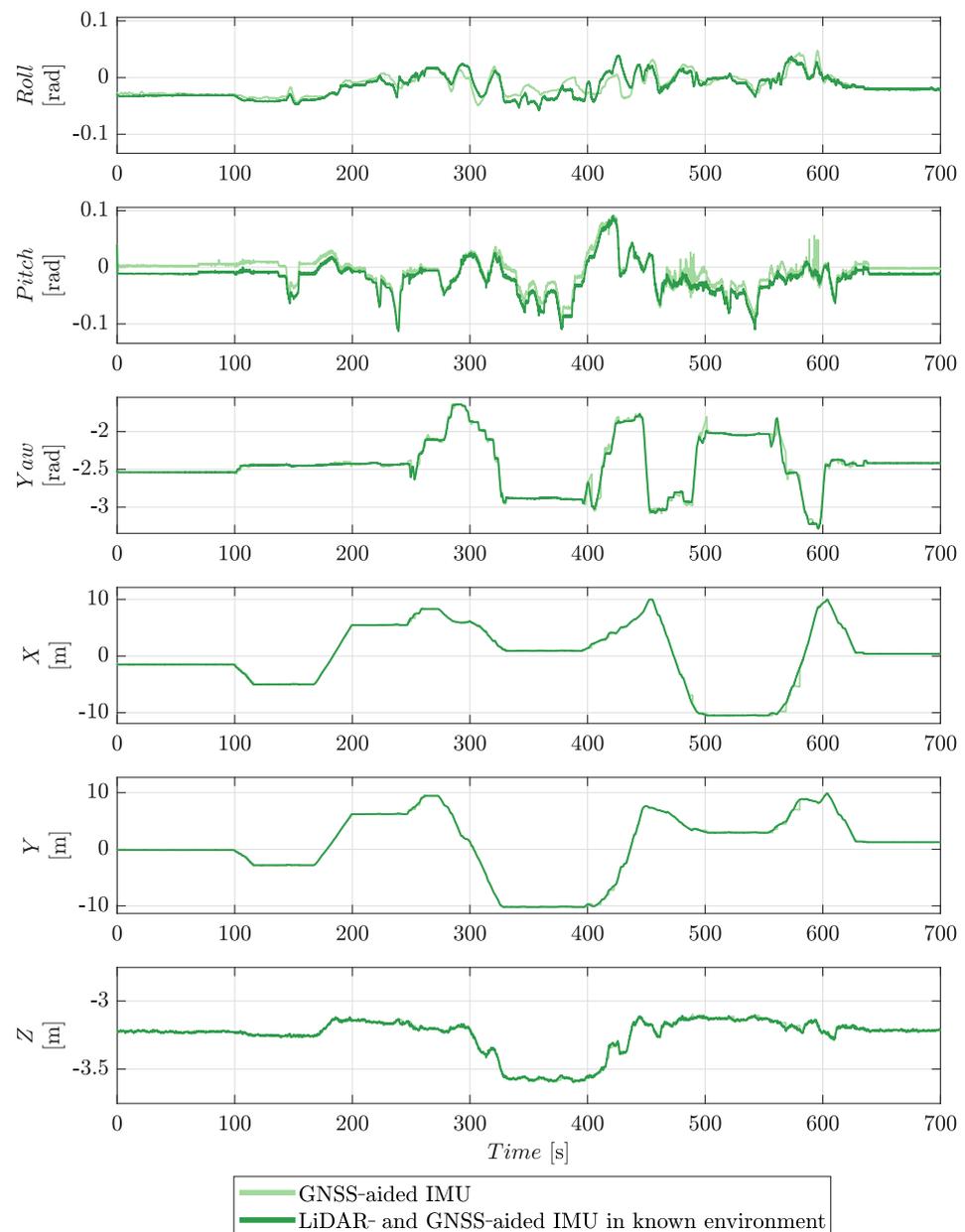


Figure 11. A comparison of the results from the GNSS-aided IMU and the LiDAR- and GNSS-aided IMU in a known environment navigation solutions.

9. LiDAR Measurement Model in an Unknown Environment

The capability to estimate pose when the environment is known helps to establish the potential performance of the LiDAR- and GNSS-aiding estimator, but is of limited practical usefulness. Most environments of interest in which this platform operates have unknown geometry. The estimator is extended to achieve the benefits of pitch stabilisation provided by the LiDAR measurement by concurrently estimating the pose and the local terrain.

For this purpose, the terrain is modelled as a height grid that divides the (x, y) plane into a regular grid. Each grid element, known as a cell, is assigned a height. The points defined by each cell in the height grid can be triangulated to develop a representation of the terrain as a surface. A representation of the triangulated height grid with states $h_{0,0}$ to $h_{3,3}$ is shown in Figure 12.

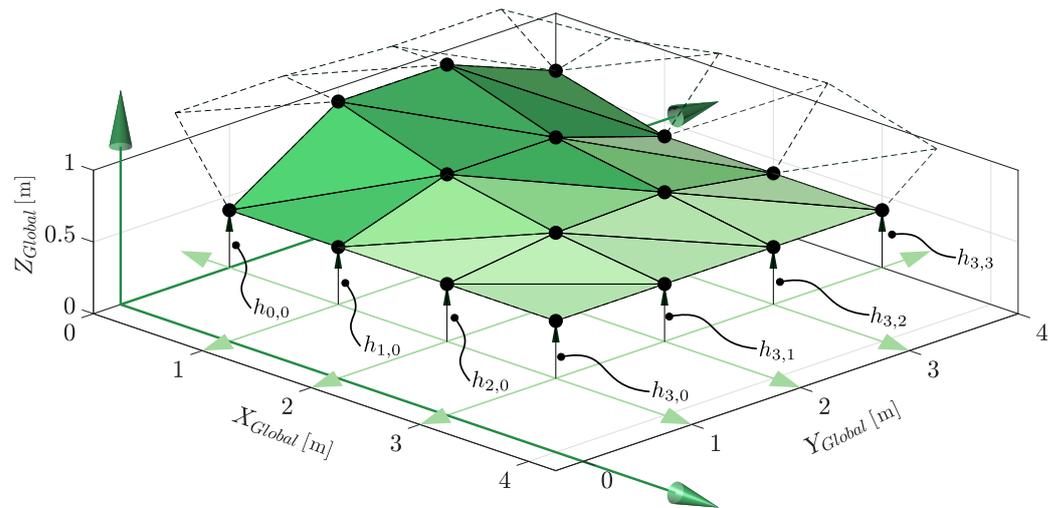


Figure 12. The terrain estimate is represented by a triangulated height grid structure. Each cell height is a state which we estimate in the filter.

Each of the $o \cdot p = q$ cell heights in the grid can be considered additional states for which estimates are sought. These states, collectively denoted x_t , augment the state vector defined by Equation (1), shown here as x_p . Here, x_t takes the form

$$x_t = [h_{0,0}, \dots, h_{i,j}, \dots, h_{n,m}]^T. \quad (73)$$

The subscript t indicates terrain. The additional terrain state estimates are appended to the state vector as follows:

$$x = \begin{bmatrix} x_p \\ x_t \end{bmatrix}. \quad (74)$$

As the terrain model is now part of the state vector, the LiDAR observation function is of the form

$$z_L = h_L(x) + v_{z,L}. \quad (75)$$

The measurement function used to estimate the terrain and pose is identical to that of the previous section. However, expected range measurements are now determined through ray-casting onto the triangulated height grid model constructed from the x_t vector, not a provided static model. The z -value or height of the vertices of each triangle of the grid, α_j , β_j and γ_j , are now defined by the elements of x_t . The measurement Jacobian for the terrain states is determined analytically from the partial derivative of the LiDAR range measurement with respect to the z -value of the points of the intercepted triangle (α_j , β_j and γ_j).

Figure 13 shows that the concurrent terrain and pose estimation system performs slightly worse than the LiDAR- and GNSS-aided IMU solution and significantly better than the GNSS-aided IMU solution. The mean point-to-model error for the concurrent terrain and pose estimation system is 0.14 m. The zero point-to-model error bin is the largest bin, with the error distribution extending to higher point-to-model error values.

Figure 14 shows the environment model generated by the concurrent terrain and pose estimation system. The terrain model is coloured based on the difference between the estimated terrain and the surveyed ground truth model. The terrain estimate shows close agreement with the truth model: generally, state errors are less than 0.1 m but extend out to 0.25 m. The terrain estimate has an RMS error of 0.126 m, which was achieved using a 1 m height grid terrain representation. Generally, areas of large error are present in high gradient locations of the terrain model. These can be difficult to capture using a regular grid terrain representation.

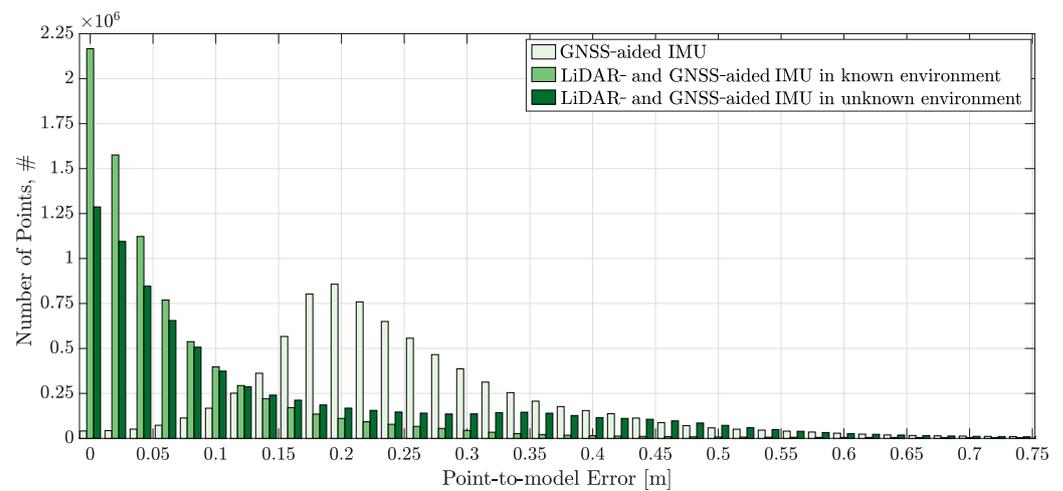


Figure 13. The point-to-model error distributions obtained by comparing the LiDAR endpoints with the ground truth terrain model from the three navigation solutions presented in this paper.

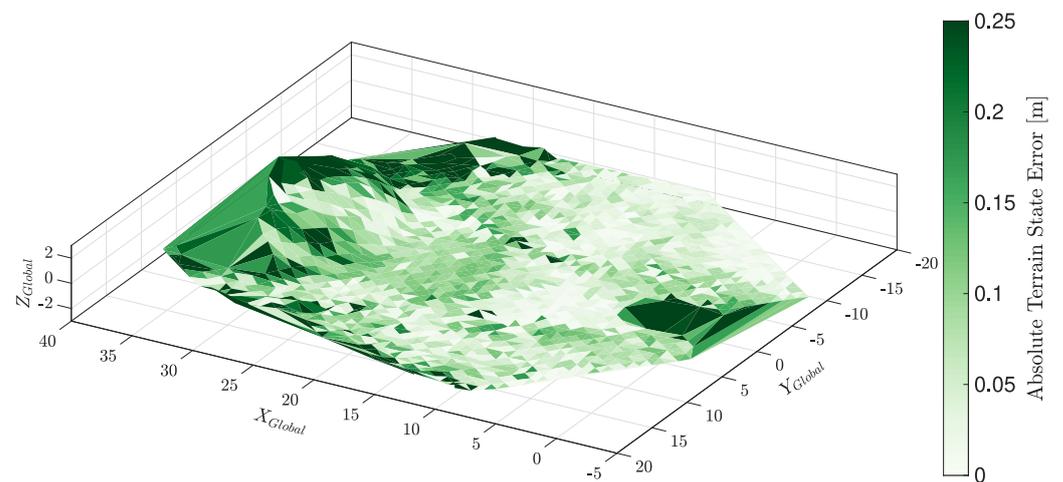


Figure 14. The error of the triangulated state mesh constructed whilst localising the platform. Here, the error of the terrain model is generally less than 0.05 m; however, there are some regions of larger error. The terrain estimates pictured here are fit for the purpose of terrain mapping in the use case this paper targets.

The estimates of the six degrees of freedom of the platform's pose are shown in Figure 15. The solution generated without *a priori* knowledge of this environment shows some deviation in both roll and pitch when compared to that obtained using a known environment. The forward-looking orientation of the LiDAR sensor clearly stabilises the pitch estimate; the roll estimate is similar to the GNSS-aided IMU solution. A wider three-dimensional scan is expected to yield better stabilisation of roll. Equally, better roll estimates could be achieved by increasing the space between the GNSS antennas. The yaw, x , y , and z degrees of freedom are similar to those estimated when the environment is known. Overall, the results demonstrate the effectiveness of the approach.

The concurrent terrain and pose estimation filter, again, required more computing per timestep. It was demonstrated to run at ~ 50 Hz on the target hardware (see Section 8).

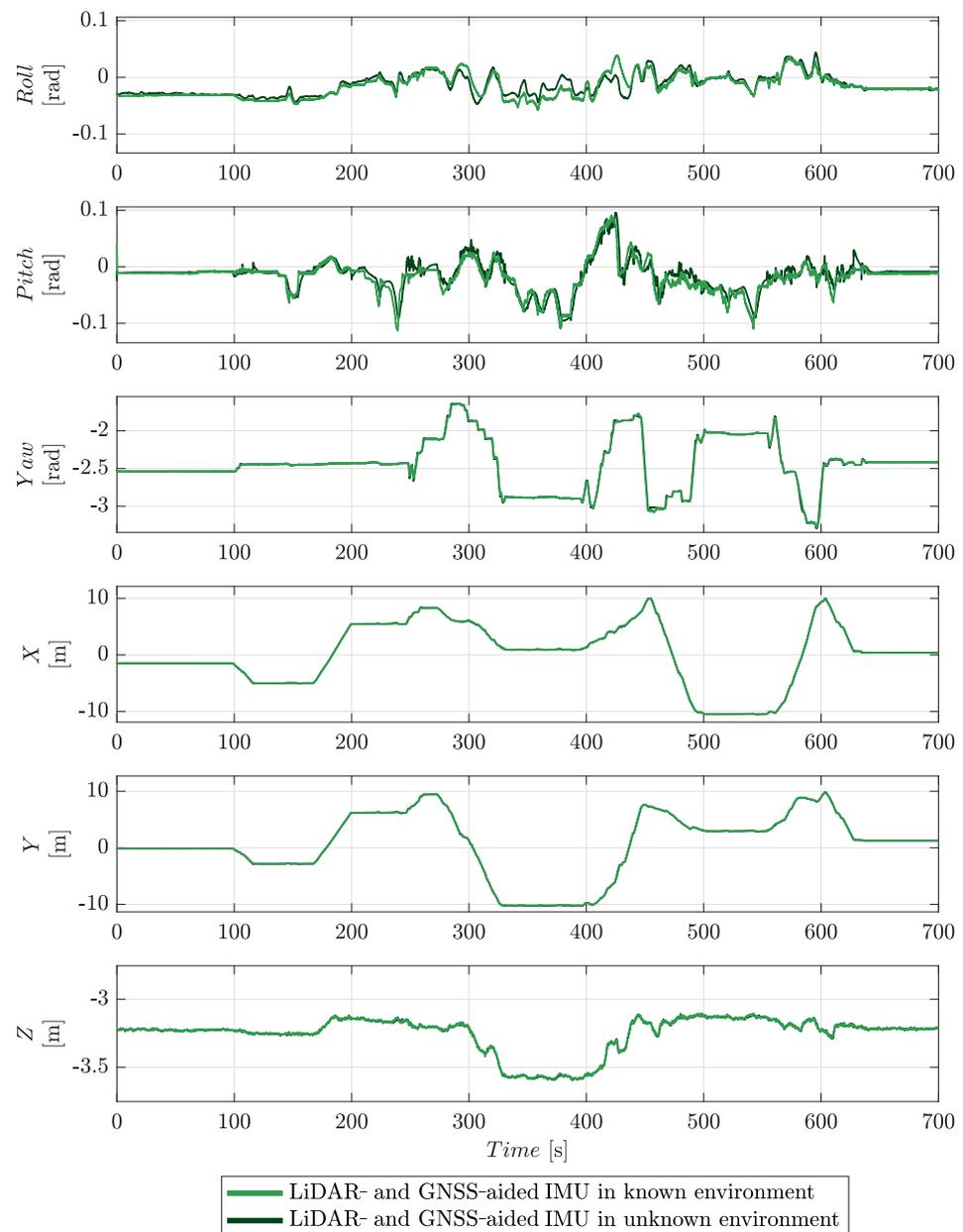


Figure 15. A comparison of the results from the LiDAR- and GNSS-aided IMU navigation solution both with and without an *a priori* environment model.

10. Conclusions

The main result from this work is to show how LiDAR can be used to stabilise two-antenna GNSS-aided IMU pose estimators in environments with known and unknown geometry. The formulation is structured to accommodate multi-rate and non-synchronous measurements. It enables continuity through short-term periods of lost measurements from one of the sensors. Lost measurements are a characteristic observed on some brands of RTK-GNSS receivers, and the solution described in this paper adapts well to this situation.

We have not explored the possibility of using LiDAR stabilisation in GNSS-denied environments; however, the approach described here could be readily adapted to such situations.

In the paper introduction, we identified that LiDAR- and GNSS-aided IMU solutions can be effective when three-dimensional LiDAR measurements are available, but that three GNSS receivers are impractical or cost-prohibitive. Our recommendation, based on experience, is that if three receivers are possible, they should be used, particularly if an accurate pose solution to high frequencies is required.

Author Contributions: Conceptualization, T.D. and P.M.; methodology, T.D.; software, T.D. and T.P.; validation, T.D.; formal analysis, T.D. and P.M.; investigation, T.D. and T.P.; resources, T.D.; data curation, T.D.; writing—original draft preparation, T.D.; writing—review and editing, P.M. and T.P.; visualization, T.D.; supervision, P.M.; project administration, P.M.; funding acquisition, P.M. All authors have read and agreed to the published version of the manuscript.

Funding: TAD is financially supported by a postgraduate scholarship funded by Australian Coal Association (ACARP) project C27063.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: The authors would like to thank Richard Hensel and Peter Beasley who conducted the collection of data used in this paper. Access to the Caterpillar D11T Bulldozer platform was facilitated by Caterpillar product distributor WesTrac Pty. Ltd. We acknowledge the outstanding support of Caterpillar Inc. who are a partner in ACARP project C27063.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

DC	Direct current, referring to the zero-frequency component of a signal.
DOF	Degree(s)-of-freedom
EKF	Extended Kalman filter
EIF	Extended information filter
FIM	Fisher information matrix
FIV	Fisher information vector
GNSS	Global Navigation Satellite System
IEKF	Indirect extended Kalman filter
IMU	Inertial measurement unit
LiDAR	Light detection and ranging
RMS	Root mean squared
RTK	Real-time Kinematic
TTT	Track-type tractor

References

1. Bryson, A., Jr.; Henrikson, L. Estimation using sampled data containing sequentially correlated noise. *J. Spacecr. Rocket.* **1968**, *5*, 662–665. [[CrossRef](#)]
2. Lefferts, E.J.; Markley, F.L.; Shuster, M.D. Kalman filtering for spacecraft attitude estimation. *J. Guid. Control. Dyn.* **1982**, *5*, 417–429. [[CrossRef](#)]
3. Shuster, M.D.; Oh, S.D. Three-axis attitude determination from vector observations. *J. Guid. Control.* **1981**, *4*, 70–77. [[CrossRef](#)]
4. Li, Y.; Efatmaneshnik, M.; Dempster, A.G. Attitude determination by integration of MEMS inertial sensors and GPS for autonomous agriculture applications. *GPS Solut.* **2012**, *16*, 41–52. [[CrossRef](#)]
5. Lim, J.; Choi, K.; Cho, J.; Lee, H. Integration of GPS and monocular vision for land vehicle navigation in urban area. *Int. J. Automot. Technol.* **2017**, *18*, 345–356. [[CrossRef](#)]
6. Gakne, P.V.; O’Keefe, K. Tightly-coupled GNSS/vision using a sky-pointing camera for vehicle navigation in urban areas. *Sensors* **2018**, *18*, 1244. [[CrossRef](#)]
7. Petritoli, E.; Giagnacovo, T.; Leccese, F. Lightweight GNSS/IRS integrated navigation system for UAV vehicles. In Proceedings of the 2014 IEEE Metrology for Aerospace (MetroAeroSpace), Benevento, Italy, 29–30 May 2014; pp. 56–61.
8. Guo, P.; Li, X.; Gui, Y.; Zhou, X.; Zhang, H.; Zhang, X. Airborne vision-aided landing navigation system for fixed-wing UAV. In Proceedings of the IEEE 2014 12th International Conference on Signal Processing (ICSP), HangZhou, China, 19–23 October 2014; pp. 1215–1220.
9. Zhang, G.; Hsu, L.T. Intelligent GNSS/INS integrated navigation system for a commercial UAV flight control system. *Aerosp. Sci. Technol.* **2018**, *80*, 368–380. [[CrossRef](#)]
10. Watanabe, Y.; Manecy, A.; Hiba, A.; Nagai, S.; Aoki, S. Vision-integrated navigation system for aircraft final approach in case of GNSS/SBAS or ILS failures. In Proceedings of the AIAA Scitech 2019 Forum, San Diego, CA, USA, 7–11 January 2019; p. 0113
11. Hecker, P.; Angermann, M.; Bestmann, U.; Dekiert, A.; Wolkow, S. Optical Aircraft Positioning for Monitoring of the Integrated Navigation System during Landing Approach. *Gyroscopy Navig.* **2019**, *10*, 216–230. [[CrossRef](#)]

12. Bhaskaran, S. Autonomous navigation for deep space missions. In Proceedings of the SpaceOps 2012, American Institute of Aeronautics and Astronautics, Stockholm, Sweden, 11–15 June 2012; p. 1267135.
13. Wang, Y.; Zheng, W.; Sun, S. X-ray pulsar-based navigation system/Sun measurement integrated navigation method for deep space explorer. *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.* **2015**, *229*, 1843–1852. [[CrossRef](#)]
14. Tweddle, B.E.; Saenz-Otero, A. Relative computer vision-based navigation for small inspection spacecraft. *J. Guid. Control. Dyn.* **2015**, *38*, 969–978. [[CrossRef](#)]
15. Gao, Z.; Mu, D.; Zhong, Y.; Gu, C. A strap-down inertial navigation/spectrum red-shift/star sensor (SINS/SRS/SS) autonomous integrated system for spacecraft navigation. *Sensors* **2018**, *18*, 2039. [[CrossRef](#)] [[PubMed](#)]
16. English, A.; Ross, P.; Ball, D.; Corke, P. Vision based guidance for robot navigation in agriculture. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 1693–1698.
17. Yin, X.; Du, J.; Noguchi, N.; Yang, T.; Jin, C. Development of autonomous navigation system for rice transplanter. *Int. J. Agric. Biol. Eng.* **2018**, *11*, 89–94. [[CrossRef](#)]
18. Lan, H.; Elsheikh, M.; Abdelfatah, W.; Wahdan, A.; El-Sheimy, N. Integrated RTK/INS navigation for precision agriculture. In Proceedings of the 32nd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2019), Miami, FL, USA, 16–20 September 2019; pp. 4076–4086.
19. Zhang, Z.; Li, P.; Zhao, S.; Lv, Z.; Du, F.; An, Y. An adaptive vision navigation algorithm in agricultural IoT system for smart agricultural robots. *CMC-Comput. Mater. Contin.* **2021**, *66*, 1043–1056. [[CrossRef](#)]
20. Lopesa, J.; Trabancoa, J. Automated control systems for civil construction machinery using RTK-GNSS: An implementation review. *Rev. Eng.* **2019**, 28–33.
21. Shcherbakov, V.; Karpik, A.; Barsuk, M. Automation of Railroad Construction Technology Using Surveying Methods. In Proceedings of the International Scientific Siberian Transport Forum, Novosibirsk, Russia, 22–27 May 2019; Springer International Publishing: Cham, Switzerland, 2019; pp. 199–208.
22. Zhang, X.; Rad, A.B.; Wong, Y.K.K. Sensor fusion of monocular cameras and laser rangefinders for line-based simultaneous localization and mapping (SLAM) tasks in autonomous mobile robots. *Sensors* **2012**, *12*, 429–452. [[CrossRef](#)]
23. Zhao, S.; Farrell, J.A. 2D LIDAR Aided INS for vehicle positioning in urban environments. In Proceedings of the 2013 IEEE International Conference on Control Applications (CCA), Hyderabad, India, 28–30 August 2013; pp. 376–381. [[CrossRef](#)]
24. Liu, S.; Atia, M.M.; Karamat, T.B.; Noureldin, A. A LiDAR-Aided Indoor Navigation System for UGVs. *J. Navig.* **2015**, *68*, 253–273. [[CrossRef](#)]
25. Núñez, P.; Vázquez-Martín, R.; del Toro, J.; Bandera, A.; Sandoval, F. Natural landmark extraction for mobile robot navigation based on an adaptive curvature estimation. *Robot. Auton. Syst.* **2008**, *56*, 247–264. [[CrossRef](#)]
26. Vázquez-Martín, R.; Núñez, P.; Bandera, A.; Sandoval, F. Curvature-Based Environment Description for Robot Navigation Using Laser Range Sensors. *Sensors* **2009**, *9*, 5894–5918. [[CrossRef](#)]
27. Im, J.H.; Im, S.H.; Jee, G.I. Vertical Corner Feature Based Precise Vehicle Localization Using 3D LIDAR in Urban Area. *Sensors* **2016**, *16*, 1268. [[CrossRef](#)]
28. Tang, J.; Chen, Y.; Niu, X.; Wang, L.; Chen, L.; Liu, J.; Shi, C.; Hyypä, J. LiDAR Scan Matching Aided Inertial Navigation System in GNSS-Denied Environments. *Sensors* **2015**, *15*, 16710–16728. [[CrossRef](#)]
29. Chen, Y.; Tang, J.; Khoramshahi, E.; Hakala, T.; Kaartinen, H.; Jaakkola, A.; Hyypä, J.; Zhu, Z.; Chen, R. Scan matching technology for forest navigation with map information. In Proceedings of the IEEE/ION PLANS 2016, Savannah, GA, USA, 11–14 April 2016; pp. 198–203.
30. Niu, X.; Yu, T.; Tang, J.; Chang, L. An online solution of LiDAR scan matching aided inertial navigation system for indoor mobile mapping. *Mob. Inf. Syst.* **2017**, *2017*, 4802159. [[CrossRef](#)]
31. Donoso, F.; Austin, K.J.; McAree, P.R. How do ICP variants perform when used for scan matching terrain point clouds? *Robot. Auton. Syst.* **2017**, *87*, 147–161. [[CrossRef](#)]
32. Chang, L.; Niu, X.; Liu, T.; Tang, J.; Qian, C. GNSS/INS/LiDAR-SLAM integrated navigation system based on graph optimization. *Remote Sens.* **2019**, *11*, 1009. [[CrossRef](#)]
33. de Haag, M.U.; Vadlamani, A.; Campbell, J.; Dickman, J. Application of laser range scanner based terrain referenced navigation systems for aircraft guidance. In Proceedings of the Third IEEE International Workshop on Electronic Design, Test and Applications (DELTA'06), Kuala Lumpur, Malaysia, 17–19 January 2006. [[CrossRef](#)]
34. Atia, M.M.; Liu, S.; Nematallah, H.; Karamat, T.B.; Noureldin, A. Integrated Indoor Navigation System for Ground Vehicles with Automatic 3-D Alignment and Position Initialization. *IEEE Trans. Veh. Technol.* **2015**, *64*, 1279–1292. [[CrossRef](#)]
35. Phillips, T.G.; McAree, P.R. An evidence-based approach to object pose estimation from LiDAR measurements in challenging environments. *J. Field Robot.* **2018**, *35*, 921–936. [[CrossRef](#)]
36. Phillips, T.; D'Adamo, T.; McAree, P. Maximum Sum of Evidence—An Evidence-Based Solution to Object Pose Estimation in Point Cloud Data. *Sensors* **2021**, *21*, 6473. [[CrossRef](#)]
37. Danezis, C.; Gikas, V. An iterative LiDAR DEM-aided algorithm for GNSS positioning in obstructed/rapidly undulating environments. *Adv. Space Res.* **2013**, *52*, 865–878. [[CrossRef](#)]
38. Yun, S.; Lee, Y.J.; Sung, S. IMU/Vision/Lidar integrated navigation system in GNSS denied environments. In Proceedings of the 2013 IEEE Aerospace Conference, Big Sky, MT, USA, 2–9 March 2013; pp. 1–10. [[CrossRef](#)]

39. Toth, C.; Grejner-Brzezinska, D.A.; Lee, Y.J. Terrain-based navigation: Trajectory recovery from LiDAR data. In Proceedings of the 2008 IEEE/ION Position, Location and Navigation Symposium, Monterey, CA, USA, 5–8 May 2008; pp. 760–765. [\[CrossRef\]](#)
40. Levinson, J.; Montemerlo, M.; Thrun, S. Map-based precision vehicle localization in urban environments. In *Robotics: Science and Systems*; Citeseer: University Park, PA, USA, 2008; Volume 3, pp. 121–128. [\[CrossRef\]](#)
41. Velodyne. *Lidar Vlp-16 User Manual and Programming Guide*; Velodyne: San Jose, CA, USA, 2016.
42. Trawny, N.; Roumeliotis, S.I. Indirect Kalman filter for 3D attitude estimation. *Univ. Minn. Dept. Comp. Sci. Eng. Tech. Rep.* **2005**, *2*.
43. González, R.; Rodriguez, F.; Guzmán, J.; Berenguel, M. Comparative study of localization techniques for mobile robots based on indirect Kalman filter. In *IFR International Symposium on Robotics*; Citeseer: University Park, PA, USA, 2009; pp. 253–258.
44. Sirtkaya, S.; Seymen, B.; Alatan, A.A. Loosely coupled Kalman filtering for fusion of Visual odometry and inertial navigation. In Proceedings of the 16th International Conference on Information Fusion, Istanbul, Turkey, 9–12 July 2013; pp. 219–226.
45. Groves, P.D. Principles of GNSS, inertial, and multisensor integrated navigation systems, 2nd edition [Book review]. *IEEE Aerosp. Electron. Syst. Mag.* **2015**, *30*, 26–27. [\[CrossRef\]](#)
46. Lee, C.G.; Dao, N.N.; Jang, S.; Kim, D.; Kim, Y.; Cho, S. Gyro drift correction for an indirect Kalman filter based sensor fusion driver. *Sensors* **2016**, *16*, 864. [\[CrossRef\]](#)
47. Bryne, T.H.; Hansen, J.M.; Rogne, R.H.; Sokolova, N.; Fossen, T.I.; Johansen, T.A. Nonlinear Observers for Integrated INS\GNSS Navigation: Implementation Aspects. *IEEE Control. Syst.* **2017**, *37*, 59–86. [\[CrossRef\]](#)
48. Koch, D.P.; Wheeler, D.O.; Beard, R.W.; McLain, T.W.; Brink, K.M. Relative multiplicative extended Kalman filter for observable GPS-denied navigation. *Int. J. Robot. Res.* **2020**, *39*, 1085–1121. [\[CrossRef\]](#)
49. Sun, W.; Wu, J.; Ding, W.; Duan, S. A Robust Indirect Kalman Filter Based on the Gradient Descent Algorithm for Attitude Estimation During Dynamic Conditions. *IEEE Access* **2020**, *8*, 96487–96494. [\[CrossRef\]](#)
50. Abdolkarimi, E.S.; Mosavi, M.R.; Rafatnia, S.; Martín, D. A Hybrid Data Fusion Approach to AI-Assisted Indirect Centralized Integrated SINS/GNSS Navigation System During GNSS Outage. *IEEE Access* **2021**, *9*, 100827–100838. [\[CrossRef\]](#)
51. Barshan, B.; Durrant-Whyte, H.F. Inertial Navigation Systems for Mobile Robots. *IEEE Trans. Robot. Autom.* **1995**, *11*, 328–342. [\[CrossRef\]](#)
52. D’Adamo, T.A.; Phillips, T.G.; McAree, P.R. Registration of three-dimensional scanning LiDAR sensors: An evaluation of model-based and model-free methods. *J. Field Robot.* **2018**, *35*, 1182–1200. [\[CrossRef\]](#)
53. Assimakis, N.; Adam, M.; Douladiris, A. Information Filter and Kalman Filter Comparison: Selection of the Faster Filter. *Int. J. Inf. Eng.* **2012**, *2*, 1–5.
54. FARO Technologies. *FARO Laser Scanner Focus3D Manual*; FARO Technologies: Lake Mary, FL, USA, 2013.