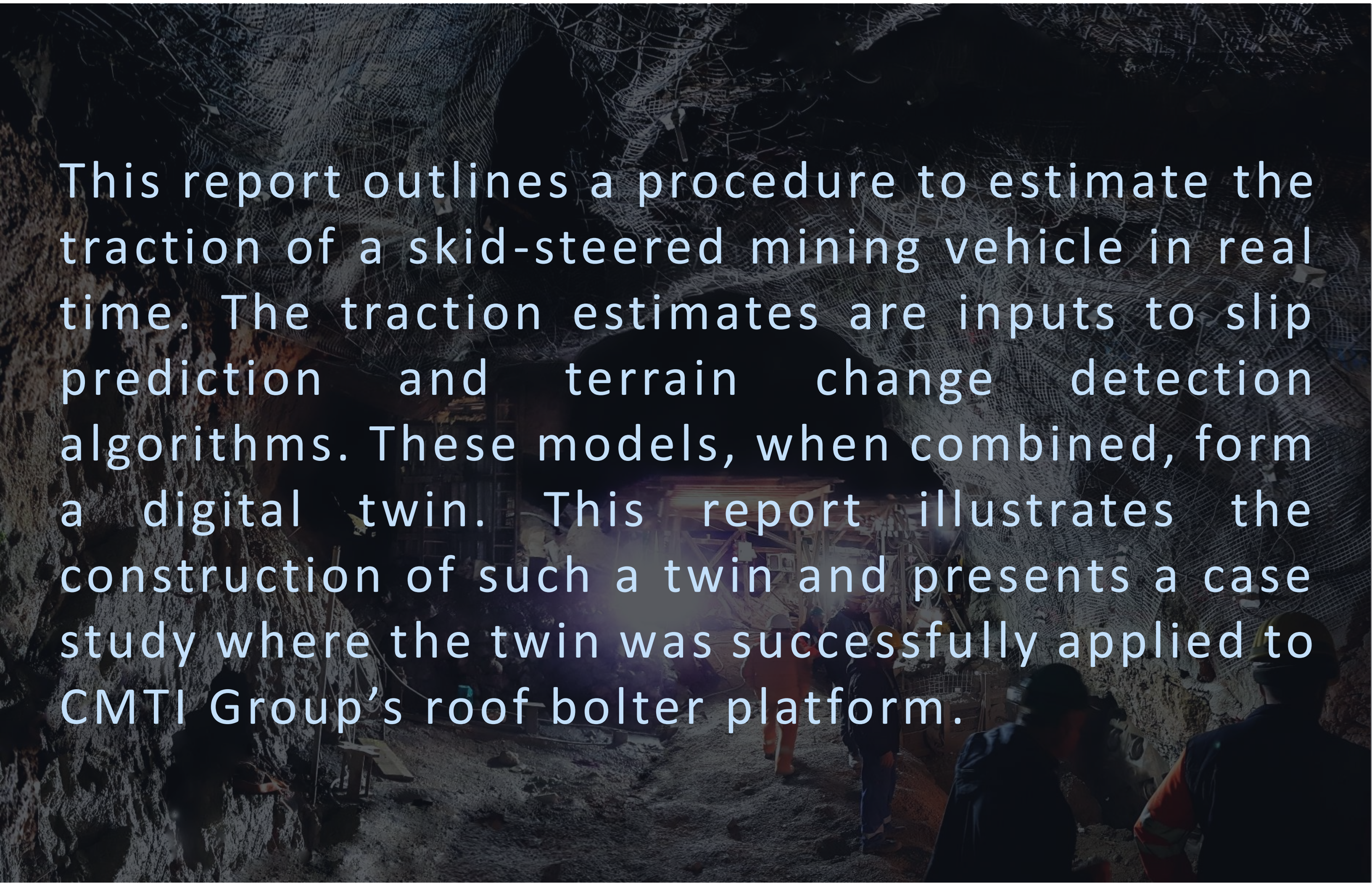
A TRACTION ESTIMATION DIGITAL TWIN
IN MECHANISED MINING

PROGRAMME	MECHANISED MINING SYSTEMS
PROGRAMME MANAGER	M PRETORIUS
RESEARCH AGENCY	UNIVERSITY OF PRETORIA
RESEARCH CENTRE	RESEARCH ACTIVITY IN MECHANISED MINING SYSTEMS (RAMMS)
DATE	27-03-2025
REVISION	1.0
AUTHOR(S)	LD VAN EYK, PS HEYNS
PROJECT TEAM	LD VAN EYK, PS HEYNS

MANDELA MINING PRECINCT
MINDS FOR MINES

AD DESTINATUM PERSEQUOR



This report outlines a procedure to estimate the traction of a skid-steered mining vehicle in real time. The traction estimates are inputs to slip prediction and terrain change detection algorithms. These models, when combined, form a digital twin. This report illustrates the construction of such a twin and presents a case study where the twin was successfully applied to CMTI Group's roof bolter platform.

1 INTRODUCTION

South African mines need to improve their ore extraction efficiency to remain competitive in the global market. Underground gold and PGM mines in South Africa face challenges like narrow veined ore reefs. This has created the need for equipment that can mine in low-profile environments to extract only the narrow ore vein. To complicate this matter, these deposits are found at difficult dip angles. For example, the UG2 orebody in the Amandelbult mining complex has an average stope width of around 1.3m and a reef dip of 18-20 degrees (van den Berg, 2000). However, Harrison (2008) indicates that these dip angles undulate, causing dip angle variations between 18-30 degrees.

These narrow reef, steep dip environments create various challenges for trackless mechanised mining machines (Jordaan, 2003; van den Berg, 2000), including:

- Machines continually running on their brakes.
- Spinning the wheel rim within the tyre due to steep dip angles.
- Sliding which ultimately leads to low-profile equipment becoming stuck against the rock face.

These factors do not only have production implications but in the case of sliding, safety implications. Harrison (2008) indicates that trials of a Sandvik XLP (extra low profile) drill rig at Amandelbult and an XLP Dozer at Union Declines were unsuccessful due to the vehicles slipping and sliding at reef dip angles between 18-30 degrees. The dip angle and associated low-profile machinery performance and safety is, therefore, a large challenge South African miners face.

The Mandela Mining Precinct (MMP) and the University of Pretoria (UP) have undertaken a research project to address these challenges under the mechanised mining systems (MMS) commercial programme. We specifically refer to

MMS24WP4, which focuses specifically on ultra-low profile machinery. To address these nascent challenges, the project proposed using a digital twin (DT) to estimate the traction between an ultra-low-profile (ULP) vehicle and the terrain it drives on. If the traction properties are better understood, operators cannot only make more informed decision when controlling a ULP platform at steep dip angles, but also make better informed decisions about machine design to mitigate possible slip events. Although the current project focuses on ULP equipment, the methodologies developed throughout the project are general enough to apply to any wheeled, skid-steered trackless mining machine (TMM).

A previous "in mechanised mining systems" report outlined a clear framework for defining digital twins. This work resulted in an open-source publication in a high-impact journal and can be found at https://doi.org/10.1016/j.cie.2024.110805. A key property of a digital twin in the proposed framework is that it creates actionable insights. That is to say, the solution must either be actionable by a human or enforceable by a control system. The traction estimates alone are not necessarily actionable and require a final mechanism to turn the traction estimate into some useful, actionable insight.

To address this requirement, we developed two algorithms. The first algorithm uses the traction estimates to predict when the vehicle may slip. This insight can be coded into a vehicle's control system, hopefully avoiding slip events. The fact that the insight is converted into a control event ensures that the current project delivers a digital twin. The second algorithm uses the traction estimates to determine when the estimated traction values start to deviate from some reference terrain. This algorithm can detect when the terrain has changed in condition, acting as an early warning system for potentially hazardous terrain. In this instance, the actionable insight is delivered to a human operator, who can make an informed decision about whether or not to continue operating.

The project was completed in conjunction with CMTI Group, which has developed a remote-controlled ULP platform that can be converted into a roof bolter, face drill rig or dozer, depending on the mounting fitted to the front of the platform. This platform was considered the only practically feasible South African platform for the purposes considered in the project. This is because CMTI had a roof bolting platform available for testing on surface. Surface testing offers a large benefit in allowing for much more cost-effective experiments before testing the final technology underground. It also ensures that no production delays are incurred due to experiments. Any other South African equipment would have to be instrumented underground, significantly increasing the required sophistication and cost of associated with underground instrumentation. This would mean very unproductive and expensive utilisation of human resources without contributing to the main objectives of the proposed project.

The current report does not necessarily report the entire project in detail, as the original milestones of the MMS24WP4 work package can be used as a reference. Rather, this report defines an "instruction manual" for creating a traction estimation digital twin, assuming one acquires the same set of instruments and applies the techniques to skid-steered TMMs.

Section 2 outlines the digital twin according to the five fundamental dimensions identified by van Eyk and Heyns (2024). The various models in this section underwent multiple iterations throughout the MMS24WP4 project, and only the final, optimised models are reported in this summary. Section 3 outlines the process that should be followed to calibrate the various sensors, models and algorithms outlined in Section 2. Section 4 describes the surface experiments conducted to gather the necessary data for twin evaluation. This data is further utilised to evaluate the ability of the twin to accurately predict the traction coefficients between the vehicle and terrain over time, the ability of the twin to predict possible slip events, and the ability of the twin to detect terrain changes. Section 5 summarises the project and accompanying conclusions. Section 6 presents recommendations to enhance the current work. Finally, Section 7 proposes the way forward from the current digital twin.

2 DIGITAL TWIN CONSTRUCTION

The main purpose of this project is to estimate the traction of a skid-steered TMM. Skid steered vehicles are quite unique, in that the wheels must slip for the machine to turn. This steering configuration allows for much tighter turning radii, which is especially advantageous in the limited space confines often experienced in underground mines. From a traction estimation perspective, the slipping behaviour can be leveraged to infer the platform's traction and motion resistance properties. If the machine turns at a slower rate for the same amount of applied power, we may infer that the terrain has higher traction coefficients, providing more resistance to turning.

Existing literature exploits this fact and correlates wheel power consumption with vehicle motion resistance and traction properties. By creating a kinematic model of a machine, one can estimate the power consumed at each wheel, subject to traction and motion resistance properties. One can optimise these properties until the difference between the two terms is as small as possible. The properties which create the smallest difference, are deemed the most likely underlying traction and motion resistance parameters between the machine’s wheels and the surface it drives on.

However, the traction estimate alone is not necessarily actionable. Therefore, the traction and motion resistance properties are used to develop two algorithms that create actionable insights, turning the traction estimation process into a digital twin. The first algorithm predicts when slip events may occur based on applied wheel torque. This insight may be programmed into the vehicle's control system to limit wheel torque, avoiding the predicted slip event.

The second algorithm alerts an operator when significant changes in the wheel-terrain interaction are present. A human can act upon this alert by inspecting the wheel and terrain conditions to determine the next steps. These insights become more powerful when implementing the terrain-change algorithms on a fleet of vehicles. If multiple vehicles detect a change between wheel-terrain interaction, it provides strong evidence that the terrain has changed, not the machine. However, if only a single machine reports differences between wheel-terrain interactions, we may be more confident that the individual machine’s wheels have changed in condition. We have not explored this option in the current project, but this is a logical extension of such technology.

The traction estimation, slip prediction and terrain-change detection algorithms can be integrated using a digital twin. A digital twin ensures that the entire actionable insight generation process is formulated in a well-defined loop between the physical asset and the digital models (the algorithms). In a previous report titled “Digital twins in mechanised mining systems”, a digital twin framework was developed and refined into an open-source article (https://doi.org/10.1016/j.cie.2024.110805). The framework presents a structured reference that defines the core dimensions, model types and properties associated with digital twins.

Figure 1 illustrates the framework, with the associated model types and properties used in constructing the traction estimation digital twin. The following sub-sections will detail the relevant models associated with each digital twin dimension and ultimately serve as the blueprint to construct a digital twin for traction estimation of skid-steered TMMs. Each section will discuss why certain model types and properties were highlighted.

2.1 PHYSICAL ENTITY

The physical entity in this project is the CMTI platform, as illustrated in Figure 2. The platform is selected as a “machine” model type according to the digital twin framework. The platform has onboard sensors that are processed and used in the twin; hence, “internal software systems” is an applicable property. Additionally, external sensors are fitted to the platform to supplement missing data, hence the “sensor” property selection.

The platform operates on the principle of skid-steer, where the vehicle steers left or right by rotating the wheels at different speeds, leading to a torque component that rotates the platform. *Table 1* summarises three parameters controlled during any steering movement: platform direction, speed and steering type. The platform has two halves, each articulating in around the longitudinal axis. This degree of freedom allows the platform to navigate uneven surfaces whilst maintaining traction at all four wheels. This is especially useful because the platform does not have a suspension.

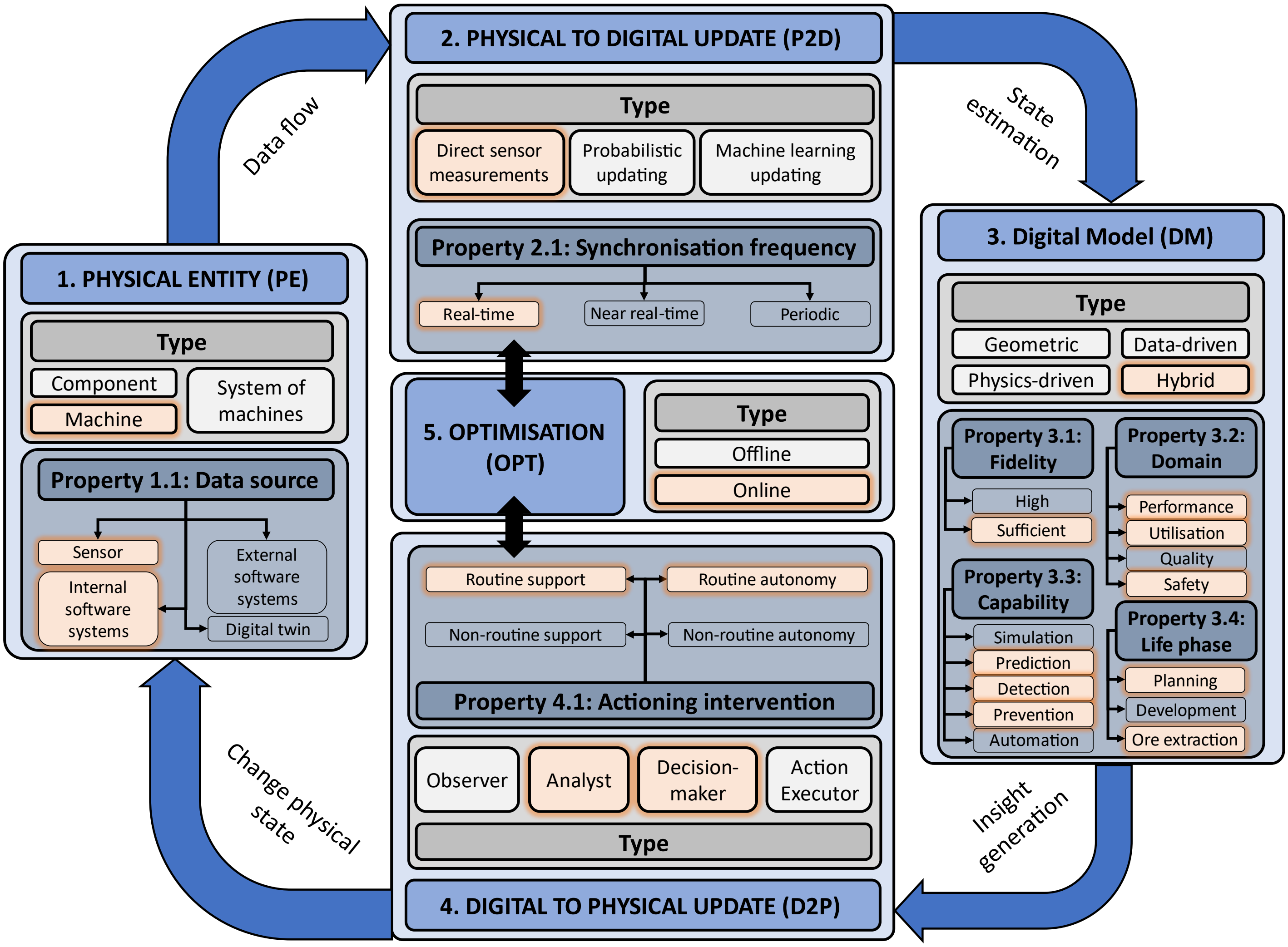


Figure 1 A high-level overview of the relevant digital twin elements (highlighted) used in this digital twin, according to the framework outlined by van Eyk and Heyns (2024).

During the literature review stage of the project, it was determined that the instantaneous centre of rotation (ICR) method was commonly applied to the kinematic modelling of skid-steered vehicles. Using this method and expanding it for power modelling, we can define a minimum set of quantities that need to be determined for the CMTI platform:

- Machine pose:
  - Rear platform roll
  - Rear platform pitch
  - Front platform roll
- Machine turning rate:
  - Yaw rate
- Machine absolute velocity (velocity relative to some fixed point, i.e. no slipping)
  - Platform longitudinal velocity
  - Platform lateral velocity
- Wheel velocities
  - Wheel speeds
  - Wheel direction
- Motor power
  - Motor current

- Motor voltage

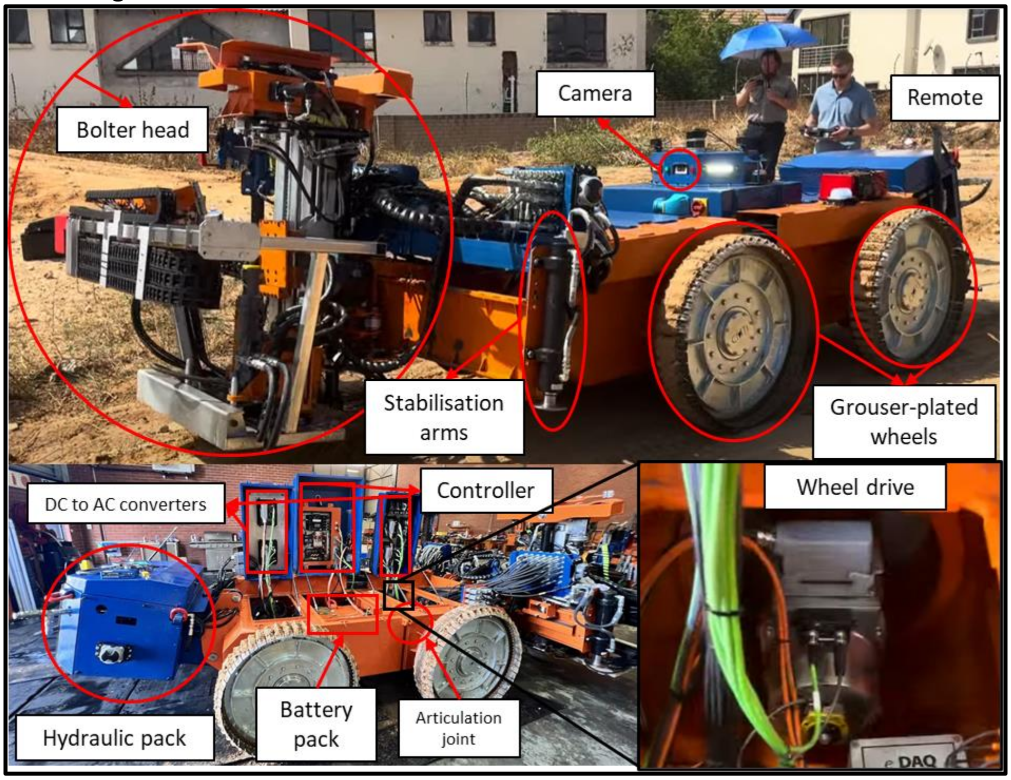


Figure 2 CMTI's roof bolter ULP platform. Key elements are highlighted for convenience of reference.

These quantities will be required for any skid-steered TMM machine using the ICR approach. Note, however, that machine pose estimates may require varying numbers of sensors depending on how many important articulating joints are present. For example, if the front and rear do not articulate, only a single roll and pitch tilt sensor is necessary. Before project implementation, there was uncertainty about the correct sensors to estimate the listed quantities. Consequently, two sensor suites were specified and simultaneously captured the same quantities. The relevant sensors used in each suite are illustrated in Table 2. Note that the wheel velocity and power estimates are measured from onboard sensors and are therefore identical across both suites.

A detailed investigation revealed that sensor suite 1, which only used a LIDAR sensor and a single-axis tilt sensor, could capture the equivalent data to sensor suite 2, which used tilt sensors, gyroscopes and ground speed sensors, at a similar, if not better accuracy[1]. This revealed that the currently proposed digital twin can be implemented with either sensor suite 1 or 2 with minimal loss in accuracy. Users can, therefore, decide whether they want a single, relatively expensive sensor (LIDAR) to measure a large array of quantities or whether a lower-cost, higher instrument count solution is more applicable.

In a related comparison, the LIDAR system was also shown to be capable of positional accuracies similar to those of a surface GPS system. The LIDAR is, therefore, useful beyond traction estimation and can be used to track traction estimates positionally in 3D space, on surface or underground. Furthermore, the LIDAR system can also map out the environment as it moves along with the platform, making it an extremely valuable sensor for autonomy underground. Due to the LIDAR's vast usefulness, the project continued with sensor suite 1, which mainly relies on the LIDAR system, and a single tilt sensor.

Figure 3 outlines a global coordinate system of the platform, along with the position of the various sensors across suite 1 and 2. Various data acquisition systems were used to capture the sensor data, illustrated in Figure 4. Most importantly, for sensor suite 1, an eDAQ XRLite was used to capture the platform's tilt and CAN bus data and a laptop to capture the LIDAR data.

---

[1] For more details about the sensor selection and comparison process, Milestone 4 should be requested and consulted.

Table 1 Summary of platform movement options

Parameter selection	Value	Meaning
Direction	Neutral	Platform should not move, brakes applied.
	Forward	Platform should move in direction of bolter head.
	Backward	Platform should move in direction of hydraulic pack.
Speed	Slow/Crawl	Wheel speed targets are lowered, allowing higher torque values to be reached.
	Fast	Wheel speed targets are increased, trading torque for faster platform movement.
Steering type	Straight	Platform will set the same rotational speed target for all four wheels in the movement direction.
	Progressive left	Right-side wheels will rotate faster than left-side wheels in the movement direction.
	Progressive right	Left-side wheels will rotate faster than right-side wheels in the movement direction.
	Skid left	Right- and left-side wheels will rotate in opposite directions. Right-hand wheels will rotate clockwise relative to right-side view observer.
	Skid right	Right- and left-side wheels will rotate in opposite directions. Right-hand wheels will rotate counter-clockwise relative to right-side view observer.

Table 2 Sensors used to build each sensor suite.

Quantity of interest	Sensor in suite 1	Sensor in suite 2
Rear platform roll	LIDAR	Roll tilt sensor 1
Rear platform pitch	LIDAR	Pitch tilt sensor 1
Front platform roll	Roll tilt sensor/linear encoder	Roll tilt sensor 2
Yaw rate	LIDAR	Yaw gyro sensor
Platform longitudinal velocity	LIDAR	Longitudinal ground speed sensor
Platform lateral velocity	LIDAR	Lateral ground speed sensor
Wheel speeds	On-board	On-board
Wheel direction	On-board	On-board
Motor current	On-board	On-board
Motor voltage	On-board	On-board

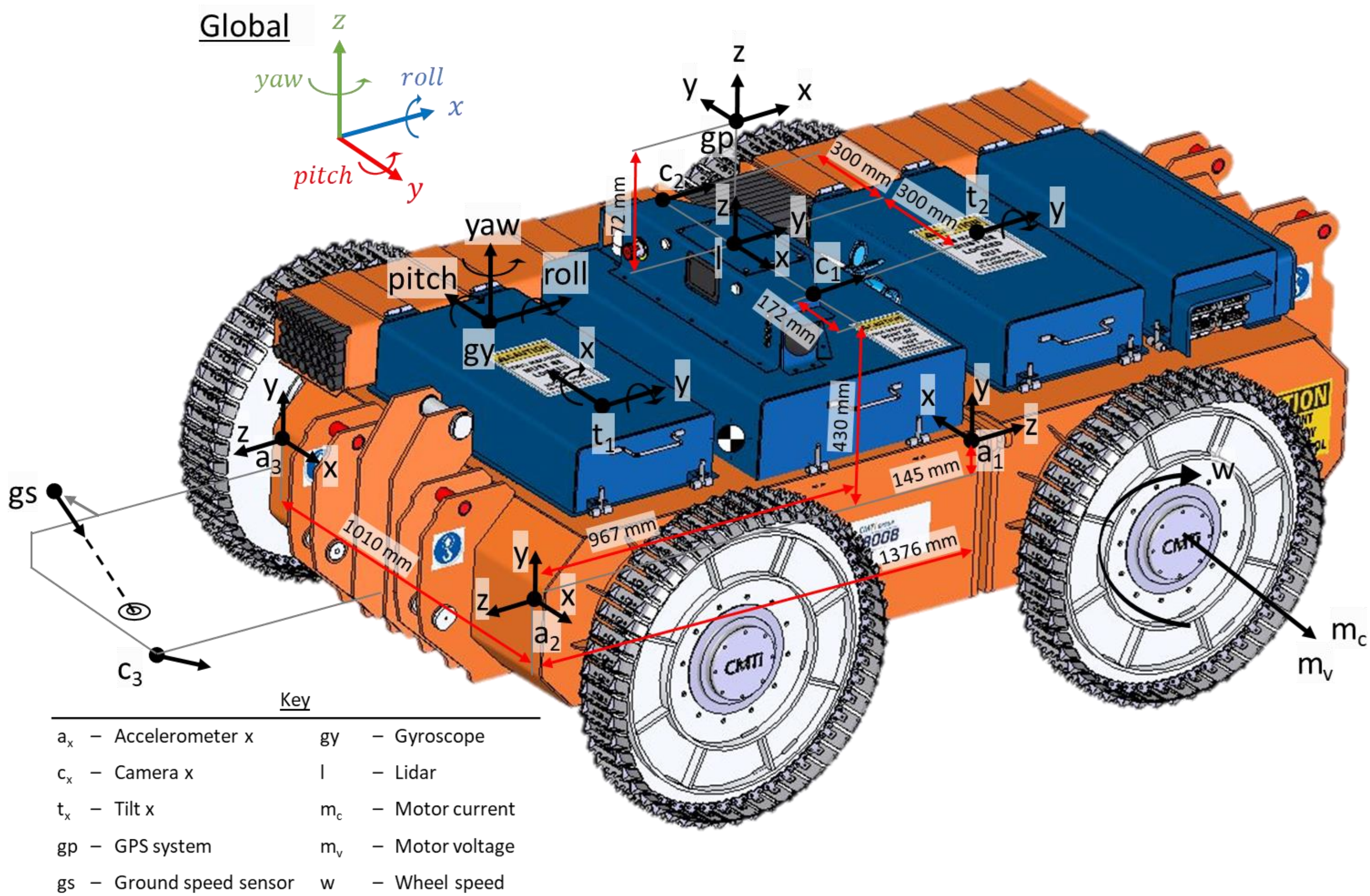


Figure 3 Schematic of various sensors and their locations. Where locations aren't precisely specified, they are deemed unimportant. The schematic is missing the rear hydraulic pack, and therefore two of the sensors (ground speed sensor and camera 3) are placed in their approximate positions, which are mounted to this hydraulic pack. Note that all axes use the right-hand sign convention. Further note that not all sensors' x,y,z axes are aligned with the global x,y,z axis.

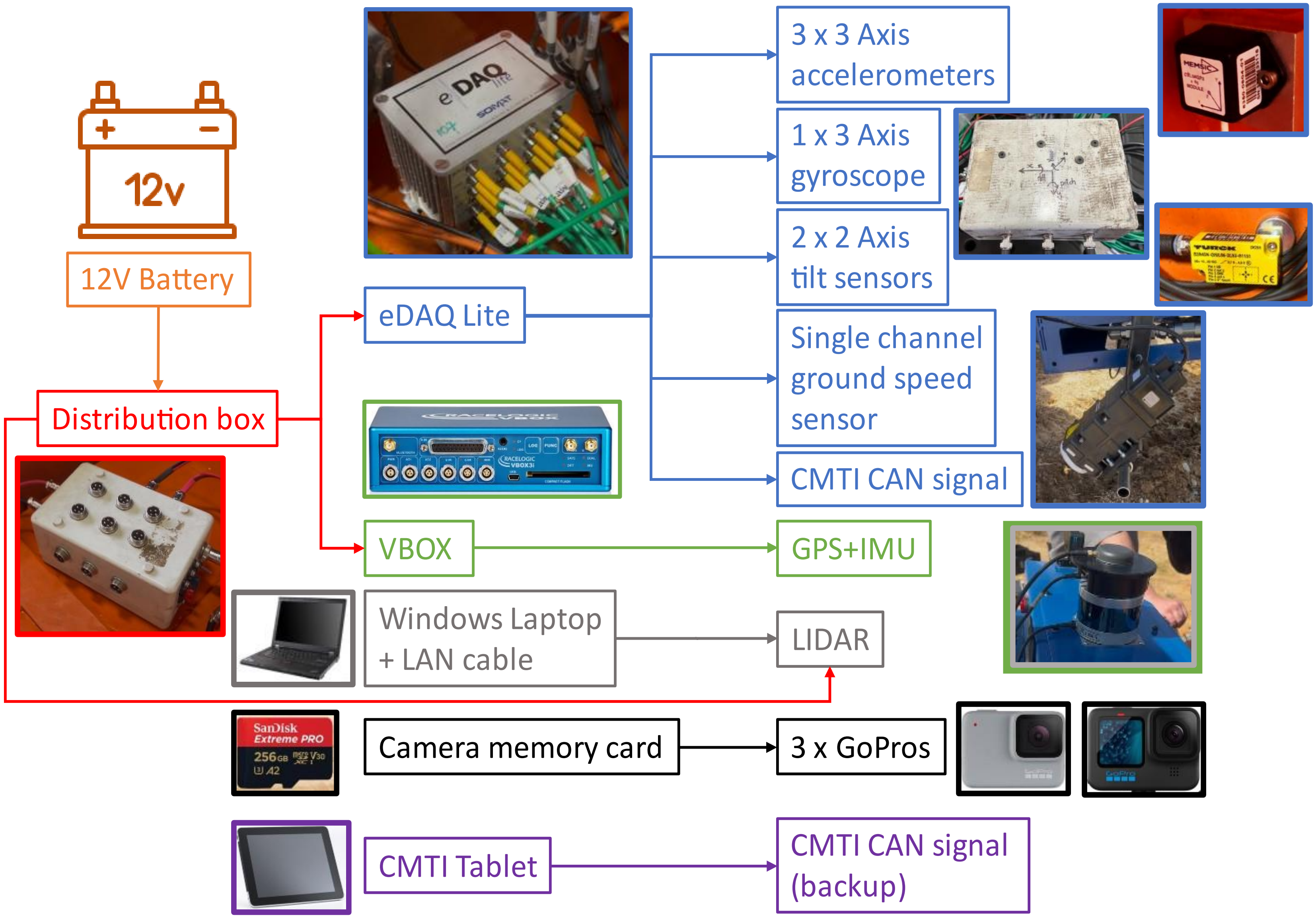


Figure 4 Schematic wiring diagram shown the data acquisition systems, their connected sensors and the power supply for each system or sensor.

2.2 PHYSICAL TO DIGITAL UPDATING

The physical-to-digital updating dimension involves converting physical states into digital states that the digital model can use. In the proposed digital twin, sensor measurements are directly used within the digital model and are therefore selected as the appropriate model type. The digital twin loop is expected to run in real-time, especially if torque control for slip prevention is implemented. Therefore, the corresponding property is selected as “real-time”.

For the most part, the physical-to-digital updating process involves converting raw sensor data into physically meaningful digital states. Although a suite of sensors was fitted to the platform, the conversion process for each sensor is not communicated in this report. Since sensor suite 1 was deemed to be the most applicable to the current project, only LIDAR and tilt sensor conversions will be discussed [2].

Due to the lack of suspension, the CMTI platform experiences mentionable vibrations while driving, and therefore, a Kalman filter was developed to remove noise from the measured data. The following sub-sections outline the conversions for the tilt and LIDAR sensors. However, before these conversions are shown, the Kalman filter algorithm is briefly discussed, as it is used in the conversion process of both sensors. The cleaned, converted output signals represent the output of the physical-to-digital updating process.

2.2.1 KALMAN FILTERING

Kalman filtering is a very broad field, and it is, therefore, impossible to capture all the nuances of the concept in a single section. Rather, key equations, their meaning and application to the current digital twin are highlighted. For a more comprehensive overview, refer to Labbe (2014).

The Kalman filter is a recursive algorithm used to estimate the state of a dynamic system from noisy observations. It operates in two main steps: prediction and correction, which systematically update estimates based on system dynamics and measurements. Essentially, the Kalman filter looks at a model of state evolution and measurements (direct or indirect) of that state and weights each contribution to get a final estimate.

2.2.1.1 Prediction step

In this step, the Kalman filter predicts the system's next state and the associated uncertainty (error covariance) based on the current state estimate and system dynamics. The future state, $\hat{x}_{k+1}^{-}$, is computed before incorporating measurements:

$$\hat{x}_{k+1}^{-} = A_d \hat{x}_k + B_d u_k \tag{1}$$

where A_d, B_d, $\hat{x}_k$ and u_k represent the state transition matrix, control input matrix, current state estimate and control input, respectively. The state transition matrix defines how the state, $\hat{x}_k$, is propagated forward in time. The control inputs, similarly, define the effect of a control input on the future state. Note that quantities with a minus sign indicate quantities that are yet to be updated with measurements. At the next step, the error covariance (P_{k+1}^{-}) is predicted according to:

$$P_{k+1}^{-} = A_d P_k A_d^T + Q \tag{2}$$

where P_k and Q represent the current error covariance estimate and process noise covariance matrix respectively. The error covariance quantifies the uncertainty associated with the current state estimate ($\hat{x}_{k+1}^{-}$). The process noise covariance matrix captures uncertainty in the system model. This uncertainty is associated with unknown dynamics like unmodelled forces or disturbances.

[2] Interested readers are once again referred to Milestone 4, which details the calibration of each sensor.

2.2.1.2 Correction step

In this step, the Kalman filter updates the predicted state estimate and error covariance using the measurement at time $k+1$. What makes the Kalman filter so powerful, is the ability to dynamically change the weighting between the state evolution model and measurements. The "tool" that determines the weighting is called the Kalman gain (K_{k+1}) defined by:

$$K_{k+1} = P_{k+1}^{-} H_d^T \left(H_d P_{k+1}^{-} H_d^T + R \right)^{-1} \tag{3}$$

where H_d and R denote the measurement matrix, and the measurement covariance matrix, respectively. The measurement matrix describes the relationship between a measured variable and the state. The measurement covariance matrix captures the uncertainty associated with these measurements. It represents uncertainty in sensor measurement associated with random noise or calibration errors. Higher values of the Kalman gain indicate higher weighting to measurements, and smaller values a higher weighting to the model.

With the Kalman gain determined, the state estimate ($\hat{x}_{k+1}$) can be updated using:

$$\hat{x}_{k+1} = \hat{x}_{k+1}^{-} + K_{k+1} \left(z_{k+1} - H_d^T + R \right)^{-1} \tag{4}$$

where z_{k+1} and $(z_{k+1} - H_d^T)$ refer to the measurement and measurement residual at time $k+1$. The measurement residual is the difference between the actual measurement and the predicted measurement. This term is weighted against the model prediction ($\hat{x}_{k+1}^{-}$) to get the final estimate for $\hat{x}_{k+1}$. This represents the best estimate of the system's state at a given timestep, accounting for both the system model and measurements.

Similarly, the error covariance (P_{k+1}) can be updated using:

$$P_{k+1} = (I - K_{k+1} H_d) P_{k+1}^{-} \tag{5}$$

where I denotes the identity matrix. The updated error covariance reflects the uncertainty in the updated state estimation. Lower values signify higher confidence in the state estimation.

This overview ignores the shapes of the matrices and vectors, which would not add much value to the understanding of the algorithm. We therefore refer interested readers to Labbe (2014). The exact parameters for the error covariance matrix (P_k), process noise error covariance (Q), measurement matrix (H) and measurement covariance matrix (R) are chosen for each sensor and will be reported in the following sections.

2.2.2 TILT SENSOR CONVERSIONS

The LIDAR sensor can only capture the rear roll and pitch of the platform. Therefore, the front roll angle of the platform is captured using a tilt sensor. The pitch angle need not be captured, as it can be determined from the rear pitch value.

The eDAQ measures the tilt sensor signal as voltage readings. The voltage and angle values have a linear relationship, with the true tilt angle (θ_{true}) being determined from:

$$\theta_{true} = \frac{V_{meas} - offset}{slope} \tag{6}$$

where V_{meas} is the measured voltage, and the slope and offset are determined through lab calibration. The lab tests used a calibrated inclinometer, where we raised an initially flat piece of wood to various inclinometer angles, and measured the voltage from the tilt sensor. This approach could determine the relationship between the tilt sensor's voltage reading and the true inclination. For the front roll tilt sensor, a slope of 0.053439 V/deg and offset voltage of 2.5V were found. Converting the measured angles into radians, one obtains the final form:

$$\theta_{x,front} = \left(\frac{V_{meas} - 2.5}{0.053439} \right) \times \frac{\pi}{180} \tag{7}$$

Figure 5 illustrates the resulting tilt readings (converted to degrees). Generally, the estimates are quite noisy (as expected) and require filtering to extract the underlying true orientation angles.

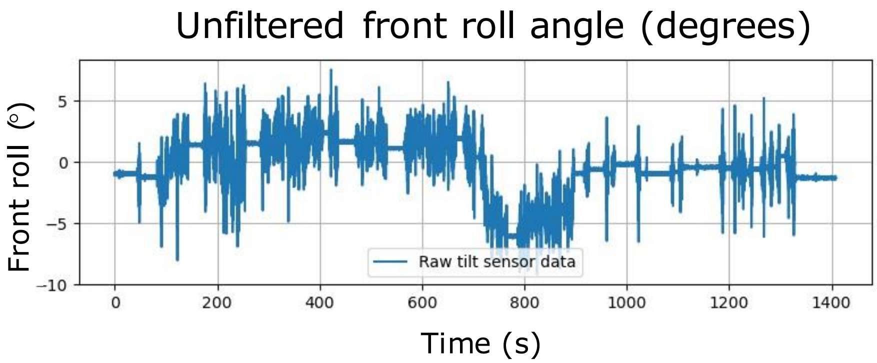


Figure 5 Tilt sensor readings (degrees) after conversion from voltage values within the eDAQ.

To counteract the noise, a Kalman filter was applied, using the following matrix values:

$$P_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{8}$$

$$Q = \begin{bmatrix} \frac{\Delta t^4}{4} & \frac{\Delta t^3}{2} \\ \frac{\Delta t^3}{2} & \Delta t^2 \end{bmatrix} \sigma_{ang-acc}^2 \tag{9}$$

$$H = [1\ 0] \tag{10}$$

$$R = \sigma_{sensor}^2 \tag{11}$$

It is difficult to estimate the initial error covariance matrix (P). Therefore, this matrix is initialised as an identity matrix because subsequent prediction and correction steps will continually update this matrix, hopefully converging to the correct values.

The Q matrix uses a timestep-dependent (Δt), constant angular acceleration model assumption. This follows the classic continuous time constant acceleration derivation, leading to the various powers of Δ_t and dependence on $\sigma_{ang-acc}$. After trial and error, $\sigma_{ang-acc} = 0.02 \frac{rad}{s^2}$ was seen to produce acceptable results. Although the error covariance matrix can adapt to the sampling frequency, we may note that the eDAQ sampled at exactly 500Hz, leading to $\Delta_t = 0.002s$. The measurement matrix indicates that only angular values are measured. Finally, the measurement noise is a scalar value (since there is only one measurement), selected as $\sigma = 0.05\ rad$.

The Kalman filter was applied to the noisy tilt data from Figure 5, resulting in filtered angles, which are illustrated in Figure 6. The resulting angular data is much cleaner yet still follows the boundaries of the measured values.

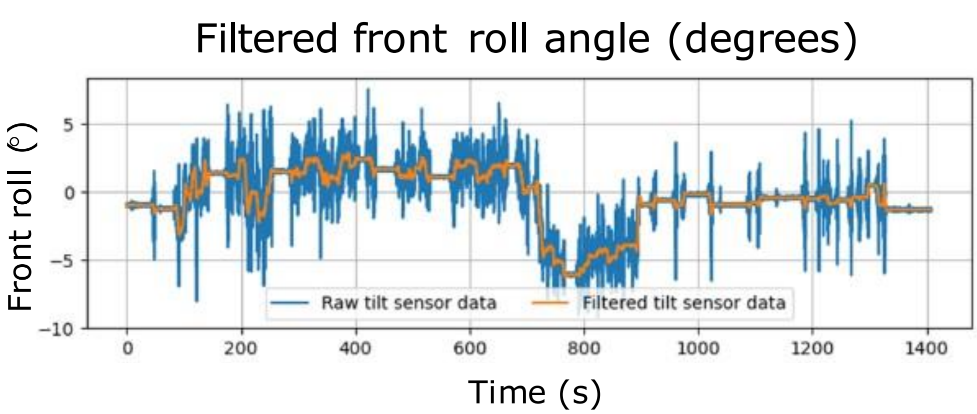


Figure 6 Kalman filtered tilt sensor data compared to the unfiltered equivalent.

2.2.3 LIDAR SENSOR CONVERSIONS

The LIDAR sensor alone does not measure all the quantities outlined in Table 2, but rather uses a simultaneous localisation and mapping (SLAM) algorithm to determine a position and rotation vector for each time-step. The SLAM algorithm localises the LIDAR sensor as it travels in three-dimensional space. To perform this localisation, the algorithm estimates all three position coordinates (x, y, z) along with the LIDAR's orientation (roll, pitch and yaw) in a global reference frame.

After applying the SLAM algorithm to the LIDAR data, estimates in the position and rotation domains are obtained. Differentiating each domain once with respect to time, gives an estimate of the platform's velocity and angular rates. These estimates will be prone to sensor noise and therefore Kalman filtering is also applied to the time-derived LIDAR data (discussed in the following sub-section).

The LIDAR data is passed through LIDARView, an open source LIDAR processing application to apply the SLAM algorithm. Although the data is post-processed in this instance, the algorithm was set up to apply the SLAM algorithm in real time, leading to identical estimates as if the algorithm was applied in real time.

The resulting estimates are reported in the global coordinate system. That is, the resulting x, y and z estimates are measured relative to a starting position of (0,0,0) with initial orientation angles of (0,0,0). This coordinate system may be useful in some circumstances, but is not required for the current project. The platform's movement must therefore be reprojected into the local coordinate frame that follows the platform, as these movement's will be comparable to the on-board sensors.

A further complexity associated with this projection format, is that initial conditions are very important. The algorithm assumes a (0,0,0) starting coordinate with the platform rotated at (0,0,0) degrees if no initial conditions are applied. It is highly unlikely that the platform will ever start a run on a perfectly horizontal plane, and therefore initial conditions need to be accounted for.

The LIDAR processing can be summarised according to three steps:

1. Correction of SLAM results with non-zero initial conditions.
2. Kalman filtering to obtain global position, velocity, angle and angular rates.
3. Reprojection to convert global quantities into the local frame.

The following sub-sections outline each of these steps.

2.2.3.1 Non-zero initial condition correction

The SLAM algorithm starts with the global position vector of (0,0,0) and global orientation vector of (0,0,0). Incorrect results would be obtained if the initial conditions differed from these assumptions. For example, if the sensor was mounted backwards, the resulting trajectory would make it seem like the platform was moving backwards, when, in reality, we neglected to account for the incorrect initial mounting.

Care was taken to ensure the initial mounting position and direction was correct. However, the initial orientation of the platform, which was certainly not (0,0,0), could not be controlled and is dependent on when a run starts logging. Therefore, post-processing code was developed that corrects the SLAM results after implementation.

The correction is applied by iterating over each row of the SLAM output and reprojecting position and orientation into the "true" initial frame. The user defines two initial pose vectors. The first is $\boldsymbol{r_{initial}} = [\text{roll}, \text{pitch yaw}]$, a 3x1 orientation vector that reads in the angular orientation of the platform in Euler angles (in degrees). The second is $\boldsymbol{p_{initial}} = [dx, dy, dz]$, a 3x1 position vector that reads in the initial position offset of the platform in metres.

After obtaining these values, the global position ($\boldsymbol{p_{SLAM}}$) and orientation ($\boldsymbol{r_{SLAM}}$) are extracted as recorded by the SLAM algorithm. All orientation vectors are converted to rotation matrices, a conversion process that is standard and therefore not presented here. This process converts the 3x1 Euler vectors into 3x3 rotation matrices that can be multiplied with other matrices to perform three-dimensional rotations. Therefore, $\boldsymbol{r_{SLAM}}$ (a vector) is converted to $\boldsymbol{R_{SLAM}}$ (a matrix) and $\boldsymbol{r_{initial}}$ (vector) to $\boldsymbol{R_{initial}}$ (matrix).

The SLAM-derived rotation matrix is premultiplied with the initial rotation matrix to correct for the sensor's actual (non-zero) initial orientation. Mathematically,

$$\boldsymbol{R_{corrected}} = \boldsymbol{R_{initial}} \times \boldsymbol{R_{SLAM}} \tag{12}$$

where $\boldsymbol{R_{corrected}}$ represents the corrected rotation matrix at each timestep. The corrected rotation matrix may be converted back to Euler angles, leading to:

$$\boldsymbol{r_{correct}} = [roll_{corrected}, pitch_{corrected}, yaw_{corrected}] \tag{13}$$

After correcting the orientation, we need to correct the positions. This is done by applying:

$$\boldsymbol{p_{corrected}} = \boldsymbol{R_{initial}} \times \boldsymbol{p_{SLAM}} + \boldsymbol{p_{initial}} \tag{14}$$

which leads to the corrected x, y and z coordinates.

For one of the surface tests, tilt sensor readings indicated that the rear half's initial roll and pitch angles were -0.2 and -7.7 degrees, respectively. The initial z-offset was estimated to be about 1m above the test site's surface. After inputting these initial offsets, the results shown in Figure 7 were obtained.

The small changes in initial conditions significantly affect the pitch and roll degrees of freedom. Furthermore, although the x and y directions aren't affected too much, the z-direction was affected significantly. The initialised results are believed to be correct, because the platform z-value keeps decreasing over time, which is sensible considering the platform started on a ramp and moved downwards towards the flat surface of the test site. Furthermore, we would like to emphasize that the initial orientation is of utmost importance, whereas the initial x,y or z offset is not. The position offsets merely aid in visual interpretation of data, whereas angular offsets affect roll, pitch and yaw angles along with position values.

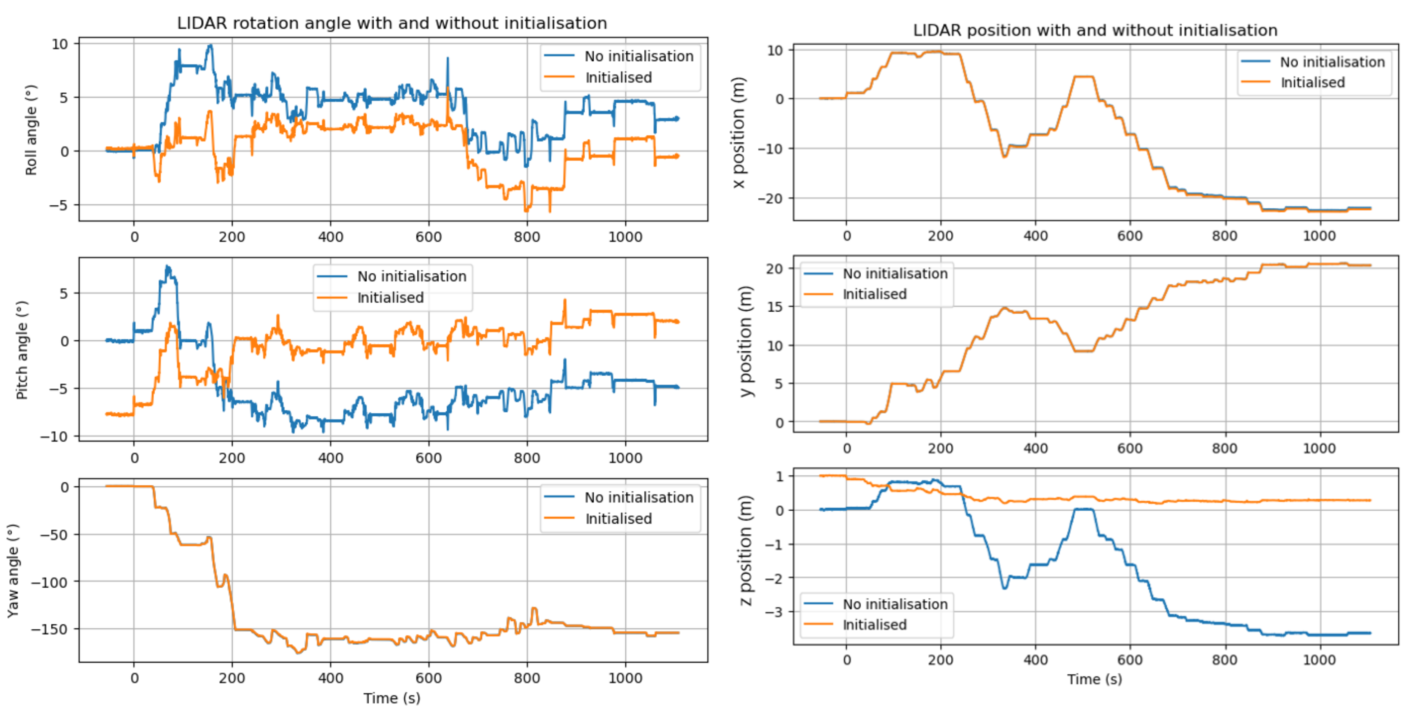


Figure 7 Left) The impact of initialisation parameters on the LIDAR system angle estimtes. We input the tilt sensor's initial readings to initialise the angular estimate of the SLAM algorithm. Right) The impact of initialisation parameters on the LIDAR system position estimates. We input an initial z-offset of 1m to initialise the position estimate of the SLAM algorithm.

2.2.3.2 Kalman filtering

Two sets of Kalman filters need to be applied to the corrected LIDAR data. The first set will clean the rotation data, as well as generate a rotation rate estimate. Rotation rate estimates are naturally obtained from the Kalman filtering process and do not have to be calculated separately.

To filter the rotations, the same Kalman filter as described earlier is used. Identical parameters are used as those of the tilt sensor, namely an angular acceleration variance of $\sigma_{ang-acc}^2 = 0.2\,\frac{rad}{s^2}$ and a sensor measurement noise variance of $\sigma = 0.05\,rad$. The resulting rotation and rotation rate measurements are shown in Figure 8.

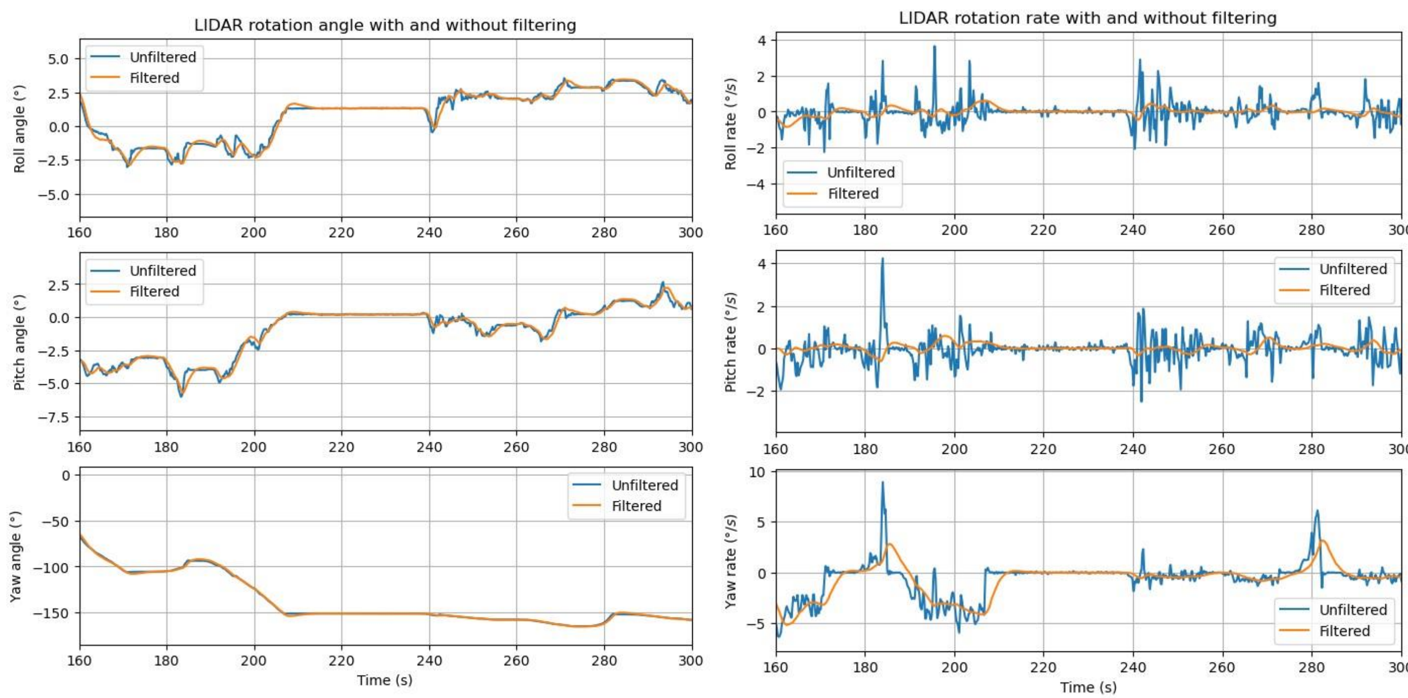


Figure 8 Left) LIDAR rotation angle estimates before and after Kalman filtering. Right) LIDAR rotation rate estimates before and after Kalman filtering.

The angle estimates were already quite noise-free before filtering but did see improvement after filtering. The angular rate values, however, saw significant improvement from the filtering, allowing the LIDAR sensor to be used as an angular velocity measurement device.

To filter the position data, an altered Kalman filter is implemented that focuses on position and velocity data. However, it is important to recognise that the dynamic equations do not change whether we convert from rotation to rotation rate data, or from position to velocity data.

Therefore, we can readily employ the same structure Kalman filter to convert positions into velocities, as we did from angles to angular rates and vice versa. It is, therefore, unnecessary to repeat the Kalman filtering formulation here, as it is identical to the Kalman filter described earlier, with the only difference being variable names. The covariance matrices' shape and structure are also identical, and are therefore not repeated. However, the variance values have been altered to create visually correct results, namely an acceleration variance of $\sigma_{acc}^2 = 1\,\frac{m}{s^2}$ and a sensor measurement noise variance of $\sigma = 0.4\,m$. The resulting filtered position and velocity data are illustrated in Figure 9.

The filtered position data remains quite clean, with some low amplitude noise being removed. The velocity data is significantly less noisy, with some low magnitude noise remaining after filtering.

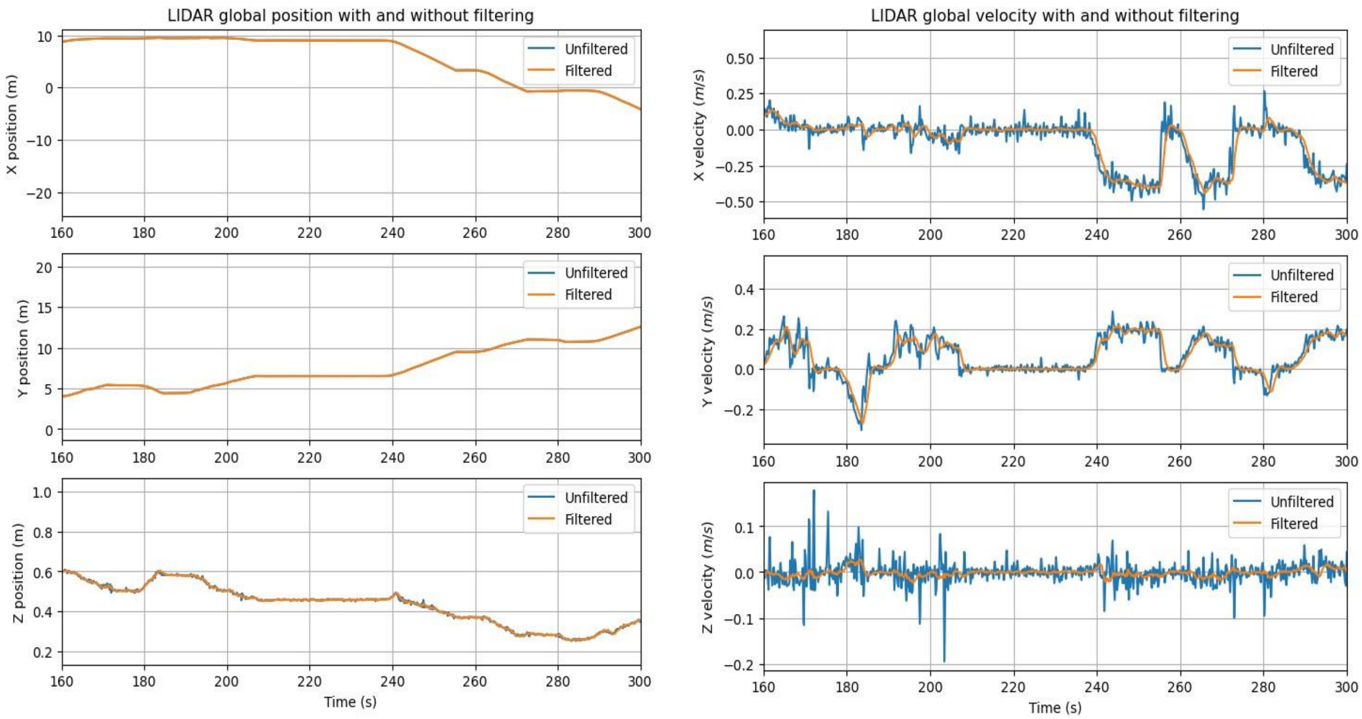


Figure 9 Left) LIDAR position estimates before and after Kalman filtering. Right) LIDAR velocity estimates before and after Kalman filtering.

2.2.3.3 Reprojecting to local (platform) reference frame

The current velocity estimates are in the global reference frame and must be projected to the platform's body reference frame, making them useful for traction estimation and slip detection.

We therefore start by constructing our global rotation matrix ($\boldsymbol{R_{global}}$) from our global corrected and filtered Euler angles ($\boldsymbol{r_{global}}$). The local rotation matrix ($\boldsymbol{R_{local}}$), is simply the inverse of the global rotation matrix. However, a property of rotation matrices is that their transpose is equal to their inverse, therefore we can write

$$\boldsymbol{R_{local}} = \boldsymbol{R_{global}^{T}} \tag{15}$$

To determine the local velocities in the platform's body frame, pre-multiply the global velocity vector ($\boldsymbol{v_{global}}$) by the local rotation matrix:

$$\boldsymbol{v_{local}} = \boldsymbol{R_{local}} \times \boldsymbol{v_{global}} \tag{16}$$

where local velocities are recorded in the same coordinate frame shown in Figure 3. The resulting comparison between the local and global velocities is shown in Figure 10.

At around 200s into the run, the direction of the platform velocity flips between the local and global projections. This is correct because the platform is initially oriented in the same direction as the global frame. However, shortly after starting the experiment, the platform skid and progressive steered by about 180 degrees, meaning velocity values become reversed with respect to the global frame.

Another clear effect that can be observed from Figure 10, is the reduction in lateral velocity amplitude. Initially, the y direction had significant (and unrealistic) velocity values, resulting from the global projection system. However, after reprojecting, much of the lateral velocity values has died away, and we are left with only lateral velocities during skid or progressive steering.

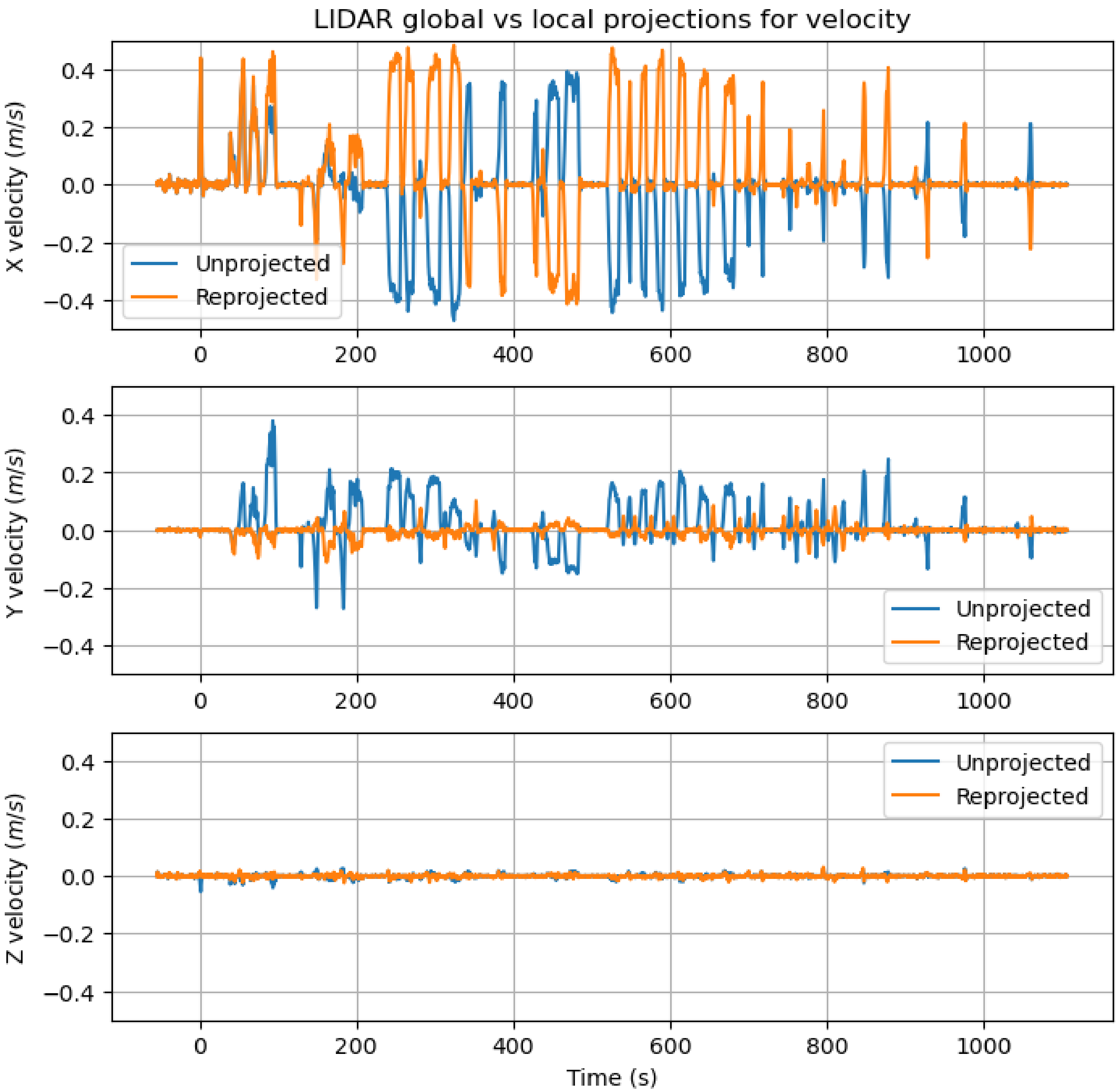


Figure 10 Impact of global to local projection on the LIDAR data.

This concludes the physical to digital updating section. The relevant sensor data has been converted and filtered so that it may be used by the digital models (next section) associated with the traction estimation task.

2.3 DIGITAL MODEL

The digital model, or in the current twin's case, models, are used to extract valuable information from the digital states. The proposed digital twin utilises five main models:

- Pose estimation model
- Force estimation model
- Traction estimation model
- Slip prediction model
- Terrain change detection model

The relationship between these models is illustrated in Figure 11.

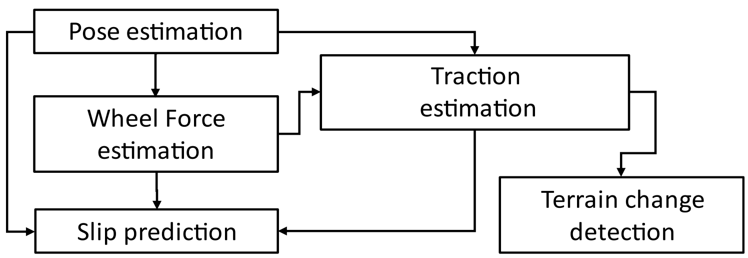


Figure 11 Five key models used in the current project, and how they link together.

The pose estimation model converts sensor data into platform pose (roll and pitch). This model, namely the Kalman filtering of angle values, has been presented in the physical to digital section.

The second and third models, associated with wheel force estimation and traction estimation, are used to convert digital states into valuable information and therefore fall under the purview of this section. The wheel force model requires pose estimation inputs to determine the corresponding wheel forces. Similarly, the traction estimation model relies on both platform pose and wheel forces.

Finally, the useful information needs to be turned into actionable insights, which are executed by the slip prediction and terrain change detection models, outlined in the following section (digital to physical updating). Both models require the traction estimates as input, with the slip prediction model additionally requiring pose and wheel force estimates.

2.3.1 WHEEL FORCE ESTIMATION

The wheel force estimation process is performed using Simscape multibody. This environment was chosen for two key reasons. Firstly, using material properties, Simscape can determine wheel forces for otherwise statically indeterminate problems. Therefore, Simscape represents a general environment where any vehicle's wheel forces may be analysed, ensuring the proposed digital twin can be expanded beyond the CMTI platform to any skid-steered TMM. Secondly, future work plans to leverage the Simscape environment to simulate the CMTI platform. Therefore, constructing the digital model in Simscape is the foundation of future work.

To design a physically accurate Simscape model, the platform must be divided into its fundamental mass blocks. Seven mass blocks are identified, each illustrated in Figure 12.

Using the following three assumptions, the mass and location of the COM for each block could be uniquely determined and are outlined in Table 3:

1. All blocks have uniform density.
2. CAD wheel masses are assumed to be accurate and are used as reported by CAD software.
3. The front and rear body blocks have identical density.

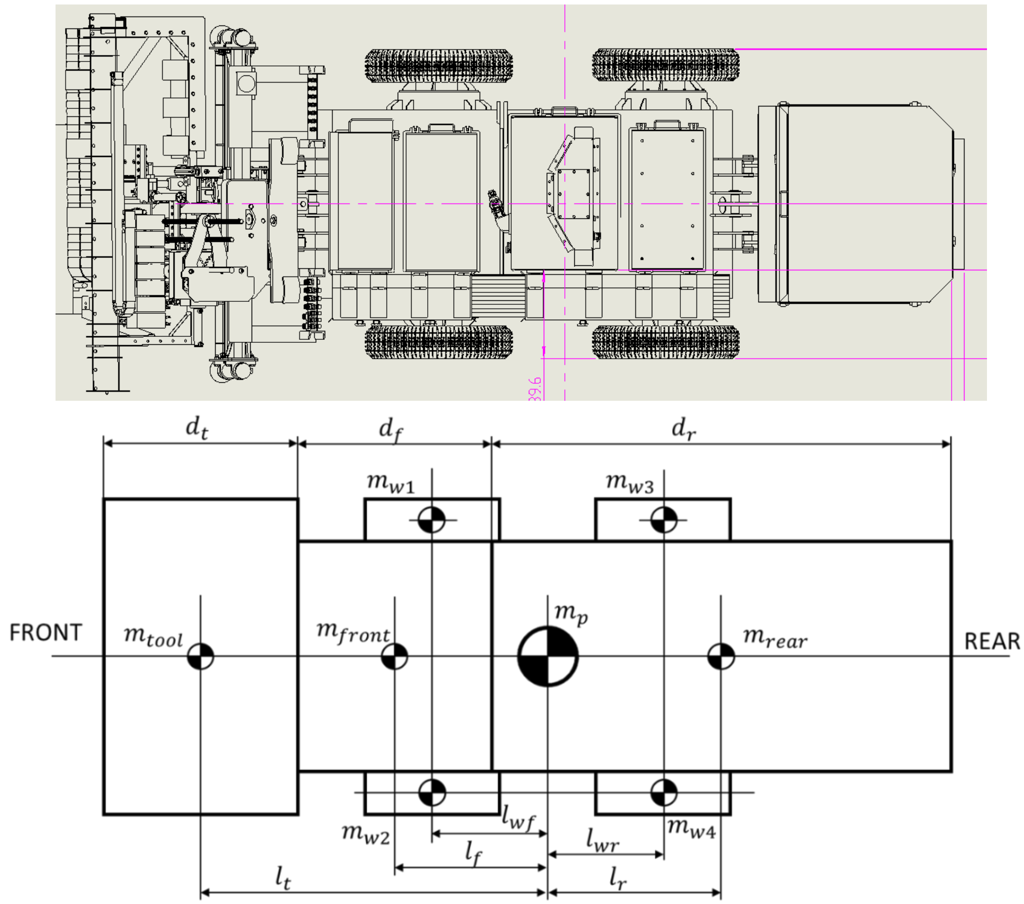


Figure 12 Top) Schematic of the CMTI platform with physically accurate proportions. Bottom) Seven mass block simplification of the platform. The large centre of mass symbol represents the global centre of mass location

Table 3 Summary of known and derived mass block values

Known values	
Quantity	Value
d_f	1260 mm
d_r	2700 mm
d_t	1400 mm
l_f	514 mm
l_r	1470 mm
l_t	1844 mm
l_{wf}	250 mm
l_{wr}	1130 mm
m_p	4612 kg
m_w	205 kg
Derived values	
m_{front}	786 kg
m_{rear}	1685 kg
m_{tool}	1320 kg

Using these mass blocks and locations, the CMTI platform could be designed within Simscape. Table 4 outlines each mass block's properties as they were read into Simscape.

Table 4 Summary of mass block properties used in Simscape

Cylindrical prismatic elements					
Mass block	Mass (kg)	Radius (mm)		Width (mm)	Density $\left(\frac{kg}{m^3}\right)$
Front left wheel	205	404		200	2000
Front right wheel	205	404		200	2000
Rear left wheel	205	404		200	2000
Rear right wheel	205	404		200	2000
Rectangular prismatic elements					
Mass block	Mass (kg)	Length (mm)	Width (mm)	Height (mm)	Density $\left(\frac{kg}{m^3}\right)$
Front body	786	1260	1400	670	665
Rear body	1685	2700	1400	670	665
Tool head	1320	1400	2000	670	704

After positioning the mass blocks and setting up the necessary joints (degrees of freedom), a physically accurate model is obtained as illustrated in Figure 13.

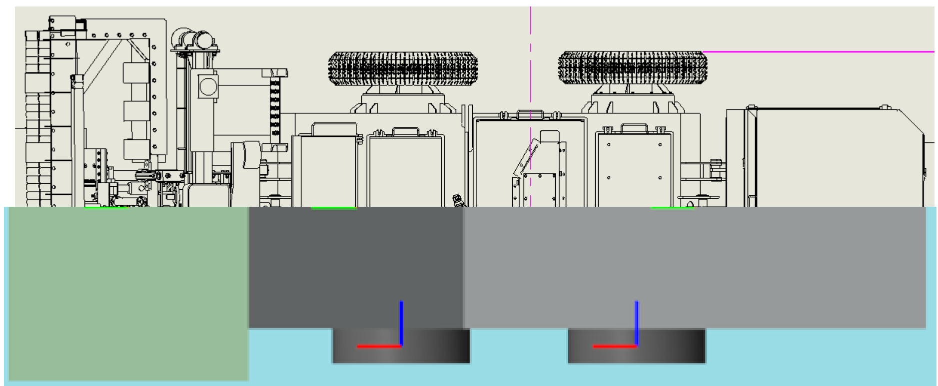


Figure 13 Comparison of CMTI platform's schematic to the Simscape model.

The platform is placed on base plates which are actuated through multiple roll and pitch angles. The platform and base plates are illustrated in Figure 14. The purpose of the actuating base plates is to obtain quasi-static wheel forces for numerous combinations of rear roll, front roll and pitch angles of the platform. Once these values are obtained, a lookup table can be created to report the wheel force for any given input roll and pitch angles. This is expected to speed up the traction estimation task, as force values can be looked up rather than simulated for each time step.

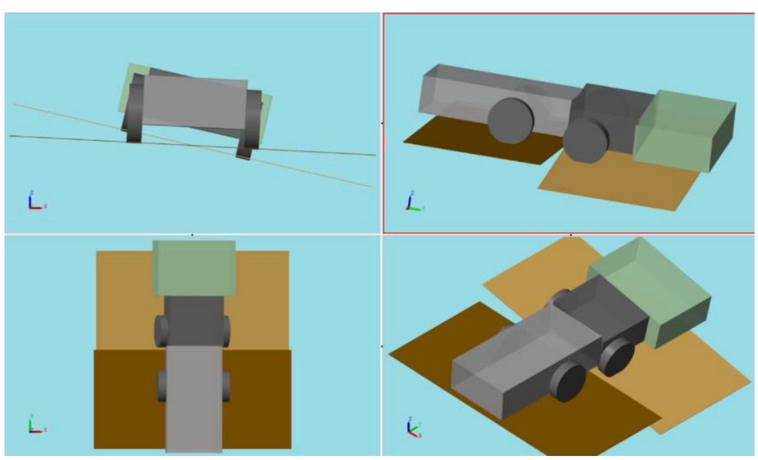


Figure 14 Illustration of platform resting on articulated bases. The platform follows the base articulation as expected.

The following section on digital twin calibration will showcase that the obtained force estimates from the Simscape model are accurate and may be trusted.

2.3.2 TRACTION ESTIMATION MODEL

The proposed traction estimation method correlates the platform's power consumption with the wheel-terrain properties. A visual summary of the flow of data in the proposed method is illustrated in Figure 15. Three main power terms must be determined and compared to the measured power. These represent the sliding traction power loss (P_s), associated with skid/progressive steering, the rolling resistance power loss (P_l), associated with the general resistance against platform motion and the platform inclination power loss (P_p), accounting for the increased effort required to work at an incline.

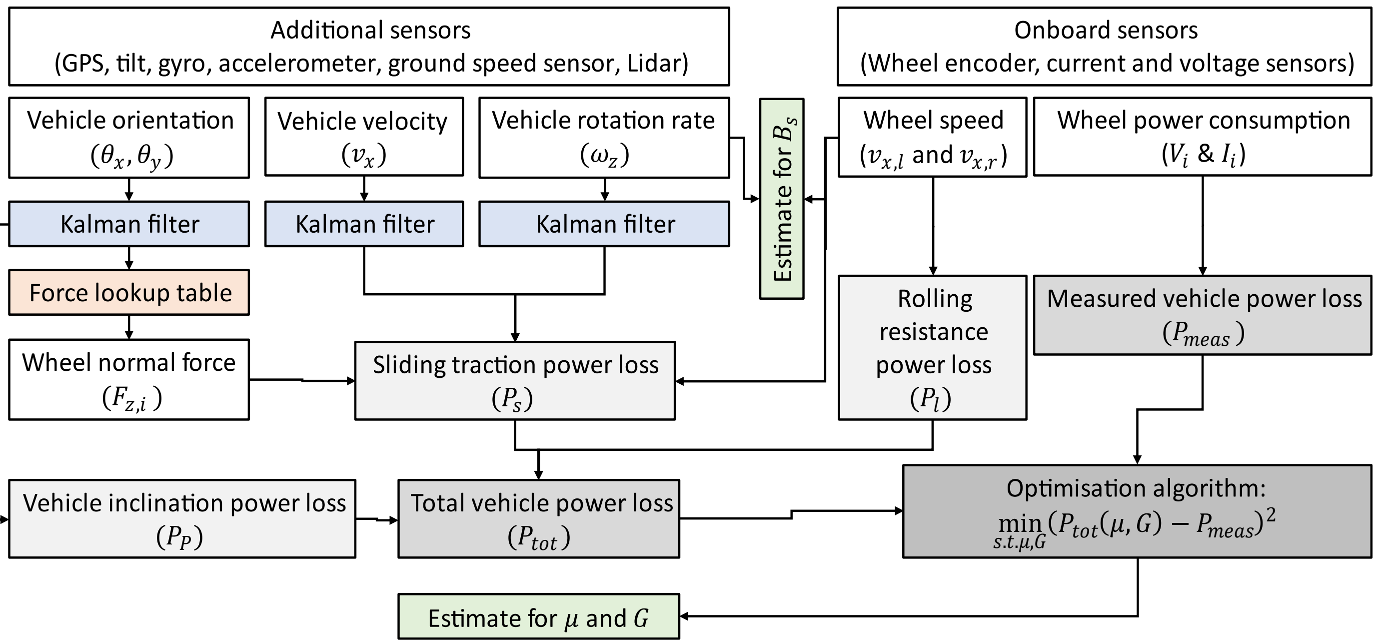


Figure 15 Schematic to show the flow of measured data into the traction estimation algorithm. Important features are coloured.

The proposed method attempts to balance the contribution of these three power terms by using traction coefficients (μ or μ_x and μ_y) and motion resistance (G) coefficients. The method requires a measurement of actual wheel power, allowing one to calibrate the modelled power values subject to the traction and motion resistance coefficients. Through this continued calibration process, an estimate of wheel-terrain traction is obtained. Figure 16 illustrates key quantities that are used to describe the ICR method.

The left and right wheel sets require different velocities to rotate by skidding. This implies that they will have some relative motion with respect to the platform's body. Therefore, for each wheel set, a point of instantaneous rotation can be calculated, labelled ICR_L and ICR_R for the left and right wheel sets, respectively. These ICRs will be unique, because each wheel set will have a unique velocity. To determine the ICR location, the average left ($v_{x,l}$) and right ($v_{x,r}$) wheel set velocities need to be determined from:

$$v_{x,l} = \frac{v_{x,fl} + v_{x,rl}}{2} \tag{17}$$

$$v_{x,r} = \frac{v_{x,fr} + v_{x,rr}}{2} \tag{18}$$

where $v_{x,fl}$, $v_{x,rl}$, $v_{x,fr}$ and $v_{x,rr}$ refer to translational velocities of the front left, rear left, front right, and rear right wheels. To determine the x and y coordinates of the ICRs, the following formulations are used:

$$y_{ICR,p} = -\frac{v_x}{\omega_z} \tag{19}$$

$$y_{ICR,l} = \frac{v_{x,l} - v_x}{\omega_z} \tag{20}$$

$$y_{ICR,r} = \frac{v_{x,r} - v_x}{\omega_z} \tag{21}$$

$$x_{ICR,p} = x_{ICR,l} = x_{ICR,r} = x_{ICR} = \frac{v_y}{\omega_z} \tag{22}$$

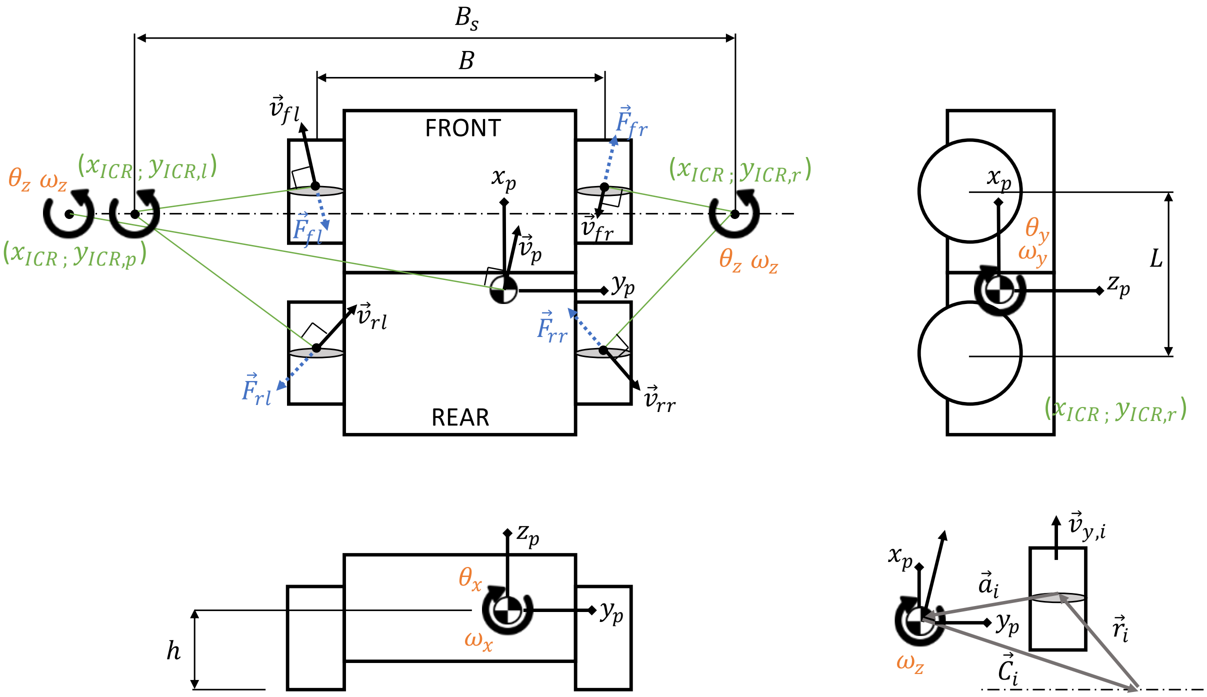


Figure 16 Reference figure indicating the key kinematic quantities for traction estimation.

Finally, a helper quantity is defined, which is shown in Figure 16 at the bottom right:

$$\vec{r}_i = \left|\vec{a}_i - \vec{C}_i\right| \tag{23}$$

where $\vec{a}$ is determined by the displacement vector from the platform COM to the wheel contact point and $\overrightarrow{C_{l,r}}$ may be determined from the ICR locations. Each power term mentioned above can be calculated from the set of kinematic quantities above. The sliding traction power term (P_s) is denoted by:

$$P_s = \mu|\omega_z| \sum_{i=1}^{4} F_{z,i} \left|\left|\vec{r}_i\right|\right| \tag{24}$$

where $F_{z,i}$ denotes the normal force acting at each of the four wheels, ω_z indicates the yaw rate of the platform and μ denotes an isotropic traction coefficient.

The rolling resistance power term is calculated from:

$$P_l = G(\left|v_{x,r}\right| + \left|v_{x,l}\right|) \tag{25}$$

where G represents the resistance coefficient, and $v_{x,r}$ and $v_{x,l}$ have already been defined as the right and left side wheel velocities. Finally, the inclination power term is calculated from:

$$P_P = mgv_x sin\theta_y \tag{26}$$

where θ_y refers to the platform's pitch angle. By combining all known sources of power loss, the formulation becomes:

$$P_{tot}(\mu, G) = \mu|\omega_z| \sum_{i=1}^{4} F_{z,i} \left|\left|\vec{r}_i\right|\right| + G(\left|v_{x,r}\right| + \left|v_{x,l}\right|) + mgv_x sin\theta_y \tag{27}$$

By measuring the actual consumed power (P_{meas}) from all four wheels, an optimization problem can be constructed:

$$\min_{s.t.\mu,G} (P_{tot}(\mu, G) - P_{meas})^2 \tag{28}$$

The goal is to estimate both unknown parameters to minimise the loss function. This process can be repeated at each time instant with new data, or over a set of N consecutive data points, representing the mean squared error over i data points:

$$\min_{s.t.\mu,G} \frac{1}{N} \sum_{i=1}^{N} \left(P_{tot,i}(\mu, G) - P_{meas,i}\right)^2 \tag{29}$$

By minimising the loss between the modelled and measured power, estimates for the isotropic traction coefficient, μ, and the terrain motion resistance coefficient, G, are obtained.

The current formulation assumes isotropic traction. However, an anisotropic traction model will be more useful if a deeper understanding of skid-steer traction is required. To the authors' knowledge, such a model does not currently exist for the current method, and therefore, the following formulation is proposed, which updates the sliding traction power loss term:

$$P_s = |\omega_z| \sum_{i=1}^{4} \frac{r_i^2 \mu_x \mu_y F_{z,i}}{\left(r_{y,i}^2 \mu_y^2 + r_{x,i}^2 \mu_x^2\right)^{0.5}} \tag{30}$$

This equation can be applied to an arbitrary number of wheels. Furthermore, this formulation assumes that the wheel-terrain traction coefficients are not dependent on the specific wheel, i. This was a valid assumption for this project, as the wheel geometry is identical across all four wheels, and the local surface is not expected to vary too much between the four wheels at any given time instant.

The current equation does not introduce new kinematic or dynamic quantities compared to the previous isotropic form, emphasizing that the proposed formulation can easily expand existing methods. Finally, if we set $\mu_x = \mu_y = \mu$, we recover the original formulation, validating that the current form is free of mathematical errors.

The novel anisotropic power loss formulation is the greatest contribution of the project, allowing traction in the longitudinal and lateral directions to be determined independently, generating greater wheel-terrain interaction insight.

Total power estimation is straightforward once pose, force and speed data have been calculated or captured. The only remaining task is to optimise the model parameters so that the calculated total platform power loss is as close as possible to the measured power values. Basic optimization schemes may be used, with the minimization problem written as:

$$\min_{s.t.\bar{\theta}} \sum_{n=1}^{N} \frac{1}{4} \sum_{i=1}^{4} \left(P_{tot,i,n}(\bar{\theta}) - P_{meas,i,n}\right)^2 \tag{31}$$

where $\bar{\theta}$ represents the unknown parameters and N represents the total number of points being optimized over.

From initial experimentation, it was found that the optimizing algorithm must be applied to groups of points based on platform movement types. For each movement type, the number of points, N, is selected and processed using the formula above. Therefore, N varies depending on how long the movement took to complete. The benefit of this approach is that a new set of parameter estimates is generated each time the platform makes a new movement. Four main movement types are considered: neutral, linear, skid and progressive steering. Each movement type requires a slightly different optimisation approach. Furthermore, some numerical instabilities affected the results, leading to additional rules being implemented during the optimisation process. These elements are described in the following sub-sections.

2.3.2.1 Neutral

No optimisation is performed during neutral periods; previous parameter values remain fixed.

2.3.2.2 Linear

Linear movements refer to periods of pure forward or reverse movements, where the left and right wheel sets move at the same velocity in the same direction. During these movements, the yaw rate (ω_z) tends to zero, leading to very large ICR values. Therefore, for these movements, set $P_s = 0$. This implies that the traction parameters are not updated and the last known traction values are assigned for this period. However, the other power terms are still applicable during linear movements. Hence, the motion resistance coefficient is optimised and updated. Linear movements represent an opportunity for the algorithm to get a "pure" estimate of the motion resistance because the traction terms do not contribute to the overall power losses.

2.3.2.3 Skid steer

Skid movements refer to periods of left or right skid manoeuvres where the left and right wheel sets move at the same velocity in the opposite direction. During these movements, the platform's lateral and longitudinal velocities are expected to remain quite small, meaning the motion resistance power term has a small contribution to the overall power loss. Traction values are expected to explain the largest amount of power loss. For this reason, the motion resistance parameter is not updated during skid movements to avoid possible numerical instabilities that may arise during small lateral or longitudinal platform velocities. However, the motion resistance power loss is included by using the last known motion resistance value. Skid steering, therefore, represents an opportunity for the traction coefficients to be updated, undisturbed by possible motion resistance contributions.

2.3.2.4 Progressive steer

Progressive steer movements refer to periods of forward left, reverse left, forward right or reverse right progressive steer manoeuvres where the left and right wheel sets move at the same velocity in the same direction but with one wheel set rotating five times slower. During these movements, the platform draws power due to a combination of motion resistance and traction effects, and therefore, both parameter sets are updated. This movement type, therefore, represents the only movement type where all parameters are simultaneously updated.

2.3.2.5 Final traction estimation optimisation algorithm

The following section discusses the order of operations required to implement the proposed traction estimation process in real-time. Naturally, the algorithm was never implemented in real-time because we needed to collect data and fine-tune the algorithm before we could prescribe the correct implementation method. However, the only difference between the project's workflow and the suggested workflow below is that we captured all the data first and then processed it. In contrast, here, we describe the simultaneous capturing and processing of data. Note that the following algorithm includes filtering steps which are only considered in the calibration section, Section 3. Therefore, to fully appreciate some of the algorithm steps to follow, read Section 3 first.

<u>Data acquisition – occurs continuously in parallel to data filtering and parameter optimisation:</u>
Implement a function that continually evaluates the movement type of the platform. Once a movement type change has been detected, trigger the data-capturing process, and follow the following steps:

1. Start logging data, capturing j sensor streams of length i, leading to a data sample $\boldsymbol{x} \in \mathbb{R}^{i \times j}$.
2. Stop recording the current data portion when the movement type changes. Start processing the newly captured portion while recording the new movement as per step 1.

<u>Algorithmic procedure – occurs in parallel to the new data portion being captured:</u>
Initialisation (run once during start-up of the system):

1. Select the loss function. We use mean squared error.
2. Select an optimisation scheme. We use Nelder-Mead.
3. Select algorithm:
    - Isotropic: $\bar{\theta} = [\mu, G]$
    - Anisotropic: $\bar{\theta} = [\mu_x, \mu_y\ G]$
4. Select initial model parameters. We use:
    - Isotropic: $\mu = 0.1$ and $G = 1$
    - Anisotropic: $\mu_x = 0.1$, $\mu_y = 0.1$, and $G = 1$

Filtering for each new sample $\boldsymbol{x}$**:**

1. If the movement type = "Neutral", skip section.
2. If length of sample less than 5s, skip section.
3. Extract latter 90% of sample, creating $\boldsymbol{x_{90}}$.
4. Calculate quantities below for each row (i) in $\boldsymbol{x_{90}}$**:**
    - $v_{x,l}$ and $v_{x,r}$
    - x_{ICR}, $y_{ICR,l}$ and $y_{ICR,r}$
    - $B_s = y_{ICR,r} - y_{ICR,l}$
    - F_z using Simscape force lookup table.
    - $\vec{r}$
    - Power model terms, subject to current $\bar{\theta}$:
        - P_s
        - P_l

        - P_p
    - Total modelled power $P_T(\bar{\theta})$
5. Filter out any rows within $\boldsymbol{x_{90}}$ where:
    - $y_{ICR,l} > 100$ or $y_{ICR,r} > 100$
    - $B_s < 0.001$

    Leaving you with the final filtered dataset, $\boldsymbol{x_f}$.
6. If no valid data points remain, skip the section.

Parameter optimisation using remaining data, $\boldsymbol{x_f}$:
Continue the following process until the loss stops decreasing or the maximum number of iterations are reached.

1. Calculate the loss between the modelled power ($P_T(\bar{\theta})$) and measured power (P_{meas})
2. Alter $\bar{\theta}$ based on the movement type to minimise the discrepancy between modelled and measured power.
    - If linear driving:
        - Only update G
    - If progressive steering:
        - Isotropic model: Update G and μ
        - Anisotropic model: Update G, μ_x and μ_y
    - If skid steering:
        - Isotropic model: Update μ
        - Anisotropic model: Update μ_x and μ_y
3. If optimisation succeeds, record the final model parameters and loss values. Otherwise, log the section as a failed optimisation and do not update the parameters.
4. Based on the current traction estimate(s), implement digital twin insights such as:
    - Slip prediction (described later)
    - Terrain change detection (described later)

These steps represent a high-level but detailed overview of the real-time traction estimation procedure developed in this project.

Referring to Figure 1, the wheel force and traction estimation digital models can be categorised into their model types and properties. The digital model dimension is a hybrid model because it fuses a data-driven model, namely the optimisation algorithm, with a physics-driven model, namely the ICR method. By fusing both methods, parameter estimates are obtained, which are informed by physics and data-driven models.

The current digital models do not have extremely high fidelity, discarding acceleration and inertial effects, and therefore have the property of sufficient fidelity. A high-fidelity property would be a suited classification for a dynamic model.

The current model addresses the domains of performance, utilisation and safety. Identifying possible slip events will lower the likelihood of uncontrollable slipping, which is a performance and safety hazard because production suffers if the machine has slipped into a position where it is stuck. Furthermore, the movement analyses that accompany the model may be used to draw in-depth utilisation insights.

The digital models possess three capabilities: prediction, detection, and prevention. The slip prediction model aims to predict available traction, which is used to prevent possible slip events. The terrain-change detection algorithm detects when significant changes in terrain (or the machine) have occurred.

The final property of the digital model is the life phase. The current digital twin is aimed at affecting both the planning and service phases of the physical asset's lifecycle. The planning phase can be influenced by varying model parameters such as mass, centre of mass location, wheel locations, etc., to determine the possible effects on slip. However, the current model is mainly aimed at the service phase, predicting slip risks and detection terrain-machine interaction changes in real-time during production.

2.4 DIGITAL TO PHYSICAL UPDATING

The digital to physical updating process converts digital model states into insights that can be acted upon by the physical entity. Upon acting on these insights, the physical entity's states are updated and the digital twin cycle starts again. For this project, two insights are generated: a prediction of when slip may occur and a detection of terrain change. The former insight may be used to limit the torque of each wheel, to avoid the slip event. Human operators may use the latter insight to aid decision-making regarding the current driving conditions on the terrain.

The following sections will outline each digital to physical updating model in more detail. However, it is useful to describe the model types and properties associated with the digital-to-physical updating process before focusing on the models.

The current project aims to analyse incoming data and decide on the insights generated from this data. The actioning intervention property of the digital twin therefore fulfils the role of an analyst and decision-maker. If CMTI implemented this twin in production, specifically the control element of the slip prediction algorithm, it would become an action executer. Furthermore, if the effort were made to visualise the data and insights of this model into a type of dashboard, the twin would also satisfy the observer role.

The digital twin has the property of providing routine autonomy. It does not require human intervention, and can automatically adjust the torque according to the current slip prediction estimates. However, the estimates are only applicable under defined scenarios. Therefore, this twin is not yet ready for non-routine environments, where, for example, a wheel is worn, a brake is broken, the platform is beached (as we will see from data in the following sections), etc.

During the testing of this algorithm, the twin will have the property of routine support, which informs the human controlling the platform that they are close to slipping, allowing the human to make a decision. The twin also warns the operator if the wheel-terrain interaction has changed.

2.4.1 SLIP PREDICTION

If an anisotropic traction model is assumed, a traction ellipse describes the envelope of possible longitudinal and lateral forces a wheel can experience before exceeding a traction limit. Figure 17 illustrates this concept.

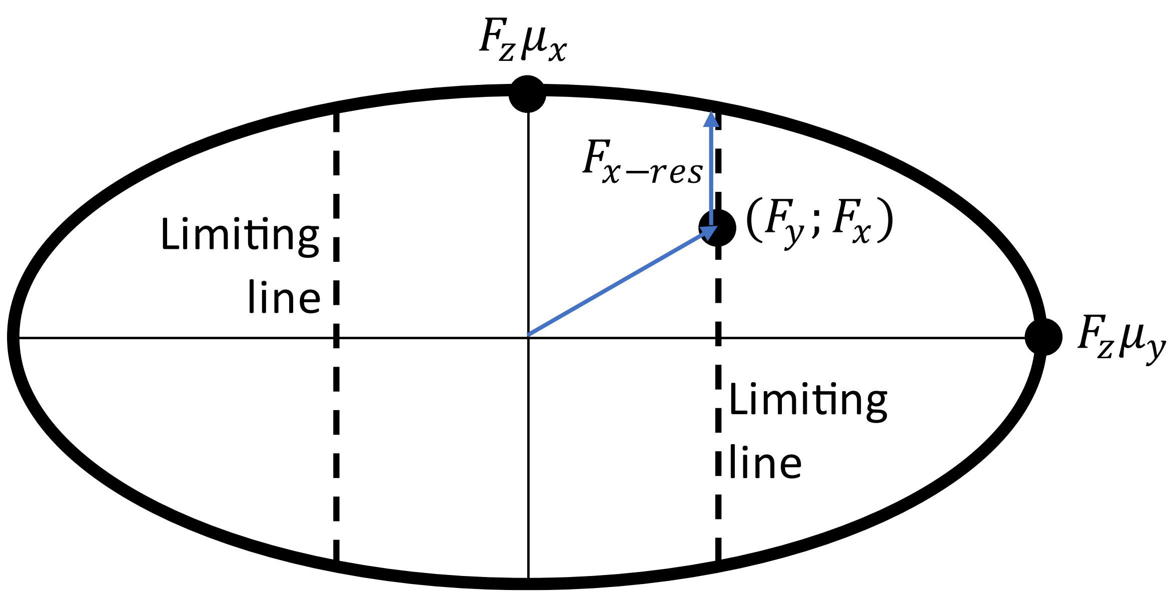


Figure 17 Traction ellipse showing the amount of residual longitudinal (x) force available for a static vehicle.

The figure shows that the longitudinal (μ_x) and lateral (μ_z) traction coefficients, along with the normal force (F_z), affect the available traction to the platform. Depending on the current roll (θ_x) and pitch (θ_y) angle of a given wheel, an offset in longitudinal (F_x) and lateral (F_z) forces will be present.

CMTI cannot control its lateral forces due to the platform's skid-steer driving style. Control can only be applied to the longitudinal forces. Therefore, we only have to account for the remaining available longitudinal force before crossing the envelope, F_{x-res}.

The following expression represents the maximum available torque, and was derived from the novel anisotropic traction assumption:

$$\tau_{max,i} = r_i F_{z,i} \left[sign(v_{x,i}) \mu_x \sqrt{1 - \left(\frac{\sin(\theta_x)}{\mu_y} \right)^2} - \sin(\theta_y) \right] \tag{32}$$

where $\tau_{max,i}$ indicates the maximum available torque to wheel i, and is a function of the wheel radius (r_i), wheel normal force (F_z), platform travel direction (positive or negative through $sign(v_{x,i})$), longitudinal traction coefficient (μ_x), lateral traction coefficient (μ_y), roll angle (θ_x) and pitch angle (θ_y).

All these variables are accessible from the traction estimation process, and therefore, this quantity can be estimated continuously as the platform drives. For this quantity to predict possible slip events, however, the true torque of the platform needs to be estimated and compared to the maximum allowable torque given here.

2.4.1.1 True torque estimation

Accurately estimating the applied torque at each wheel is difficult and would require calibrated strain gauges placed on each wheel shaft or hub. Alternatively, a correlated quantity, namely the motor current, can be used to estimate the motor torque. Although the relationship between torque and motor current is assumed to be linear, this is an oversimplification and may be inaccurate during high torque events. However, it is the best information available to us, and we shall therefore use it.

A motor constant, K_t of 3.22 Nm/A_{rms} is reported in the motor specification sheet. To use this conversion factor, the following relationship applies:

$$\tau_{est,i} = K_t \times A_{phase,i} \times 20 \tag{33}$$

where the estimated torque ($\tau_{est,i}$) for wheel i is equated to the motor phase current ($A_{phase,i}$) for wheel i multiplied by the motor constant (K_t). The CMTI platform uses a 20:1 planetary gearbox to increase wheel torque and therefore a multiplication factor of 20 is included in the current formulation.

The traction estimation process recorded wheel power values from CMTI’s onboard current and voltage sensors. Therefore, we already have access to the necessary quantities, and can easily generate real-time estimates of the applied platform torque and compare these to the maximum allowable torque from the previous section. This process is discussed in a later section.

2.4.1.2 Preventing slip events

Slip prediction and prevention are fairly straightforward once the available and measured torque values are known. We expect the platform to slip if the applied torque exceeds the available wheel torque. To avoid slipping, the platform must adhere to:

$$\tau_{est,i} < \tau_{max,i} \tag{34}$$

However, it is important to recognize that, by definition, the wheels must slip during progressive and skid steering movements. Therefore, possible slip events during these movements should not be prevented because slip is necessary and expected. However, during straight-line driving (forward or reverse), the platform must not slip and must, therefore estimate the allowable torque that the platform can apply to each wheel ($\tau_{max,i}$) and ensure that the applied torque does not exceed this value.

2.4.1.3 Warning system

The current algorithm may require some calibration before being used in a control application. A warning system is proposed to predict slip events, allowing the operator to decide whether to reduce wheel torque (or speed). Therefore, the algorithm's performance can be monitored in production environments before committing to its autonomous control system application.

To create such a warning system is fairly simple. The software must warn the user that they have entered a regime where slip is highly likely. This regime is present when:

$$\tau_{est,i} > \tau_{max,i} \tag{35}$$

Put plainly: When the estimated wheel torque value exceeds the model's estimate for the maximum available torque, the user should be warned.

2.4.2 TERRAIN CHANGE DETECTION

An unsupervised anomaly detection model can be trained to predict terrain changes by using the traction (μ or μ_x and μ_y), motion resistance (G) and slip track (B_s) parameters from the traction estimation process as terrain indicators. After some initial experimentation, simple linear classification models were shown to be insufficient for this task. We therefore opted to use a one-class support vector machine (SVM). SVMs are commonly used and are well documented. Therefore, we will not repeat the fundamental operation of an SVM here. Rather, we will present a simplified description of what the SVM model attempts to do.

The one class SVM is a model that fits a model to a reference data set. Then, given some new data, the model determines whether the new data is similar to the reference dataset (belonging to the same class) or whether it is different (doesn't belong to the same class). This approach can be leveraged to detect changes in terrain conditions.

A reference dataset can be captured (perhaps during good terrain and machine conditions), and used as the training dataset. New data can then be fed to the trained SVM and a conclusion can be made whether the new data is similar to or different from the reference training data. If the SVM classifies the data points as similar, terrain conditions (or machine) are similar. However, if the SVM identifies the data points as different and not belonging to the same class, a change in terrain or machine condition has occurred.

For the current SVM, a data point is described by the traction, motion resistance and slip track parameters at any given instant:

$$\boldsymbol{x} = [\mu_x, \mu_y, G, B_s] \tag{36}$$

For the isotropic model, the longitudinal and lateral traction coefficients are replaced by the single traction coefficient, μ.

The One-Class SVM aims to identify a function, $f(\boldsymbol{x})$, that is positive in areas where training data is present and negative in regions lacking data—these regions are likely to contain outliers not included in the training set. After the model is trained, new points can be assessed for similarity: points yield a positive classification if considered inliers and a negative classification if regarded as outliers.

In summary, the proposed one-class SVM determines whether unseen data is classified as belonging to the reference training data. If the test data is seen as an inlier, we may conclude that the terrain (or machine) has not changed. However, if the data is labelled as an outlier, we may conclude that the terrain (or machine) has changed since the reference set was captured. We may use the proportion of inliers to measure how much the terrain has changed, with the threshold ultimately determining whether the measured change is significant enough to be classified as a different terrain.

2.5 OPTIMISATION

The final digital twin dimension is the optimisation dimension, which describes the continual optimisation of the physical entity through the digital model's insights. For this digital twin, the digital state is continually updated in real-time, leading to the property of online optimisation, as illustrated in Figure 1.

3 DIGITAL TWIN CALIBRATION

The previous section outlined the various sensors, models and algorithms associated with the traction estimation digital twin. These elements need to be calibrated before digital twin implementation. This section will calibrate any remaining parameters from the various sensors, models and algorithms.

3.1 SENSOR CALIBRATION

Recall from Section 2 that seven primary sensors are required in sensor suite 1:

- Pose, angular rate and velocity:
  - Tilt sensor for front roll.
  - LIDAR for rear platform roll and pitch, yaw rate, longitudinal and lateral velocity.
- Motion sensing:
  - On-board wheel speed sensor.
  - On-board wheel direction sensor.
- Power consumption:
  - On-board motor current sensor.
  - On-board motor voltage sensor.

The tilt and LIDAR sensors do not require any further calibration, as their calibration process was described in Section 3. The on-board sensors, however, are captured as CAN signals and need to be calibrated to be physically meaningful.

3.1.1 MOTION SENSING

3.1.1.1 Wheel direction sensor

CMTI uses three wheel speed values, namely 0, 1 and 2. A zero value means the wheel motor is not turning (i.e. stationary). A value of one refers to clockwise wheel rotation, when looking at the motor from outside the platform. A value of two refers to a counter-clockwise wheel rotation from the perspective of an outside observer.

These values make the interpretation of wheel speeds difficult, and therefore we recalibrated the values to become -1 if the wheel makes the platform travel backwards, 0 if the wheel is stationary and 1 if the wheel makes the platform travel forwards. The wheel set conversions are outlined in Table 5.

Table 5 Wheel direction calibration table

Wheel	CMTI CAN Value	Recalibrated value
Front left	0	0
	1	-1
	2	1
Rear left	0	0
	1	-1
	2	1
Front right	0	0
	1	1
	2	-1
Rear right	0	0
	1	1
	2	-1

3.1.1.2 Wheel speed sensor

The wheel speed values are captured from the CAN bus and are assigned integer increments between 0 and 10. Therefore, 11 unique wheel speed values are measured. A conversion of 7.8 radians per minute per 1 CAN bus unit was used. This speed referred to the wheel speed, which is 20 times slower than the motor speed. The following conversion equation may be used to convert the CAN units (CAN_{speed}) into angular wheel speed (ω_{radps}):

$$\omega_{radps} = CAN_{speed} \times \frac{7.8}{60} \tag{37}$$

We multiply by 7.8 as before and divide by 60 to convert radians per minute into radians per second.

To obtain an estimate of the wheel speed under no-slip conditions (v_{wheel}), use:

$$v_{wheel} = r_{wheel}\omega_{radps} \tag{38}$$

where $r_{wheel} = 0.406m$.

3.1.1.3 Movement calibration.

Using the wheel direction and speed value, one can estimate the platform motion, which is necessary for the traction estimation process. The combination of wheel direction (CAN units) and speed values with the corresponding movement type is reported in Table 6.

Table 6 Platform wheel speed combinations that lead to various movement types. The current project uses a threshold of 0.15rad/s

Movement type	Front left value	Front right value	Rear left value	Rear right value	Right wheel set speed – Left wheel set speed
Neutral	0	0	0	0	N.A.
Forward	2	1	2	1	<Threshold
Backward	1	2	1	2	<Threshold
Skid right	2	2	2	2	N.A.
Skid left	1	1	1	1	N.A.
Progressive right forward	2	1	2	1	<-Threshold
Progressive left forward	2	1	2	1	>Threshold
Progressive right backward	1	2	1	2	<-Threshold
Progressive left backward	1	2	1	2	>Threshold
Any other combination is faulty data and should not be used					

3.1.2 POWER SENSING

The measured line voltage and line current CAN values were already in the correct units of Volts and Amps, respectively and therefore did not require further calibration. The only remaining conversion was to convert the current and voltage values into a power value. The CMTI platform uses three phase motors to drive the wheels. To calculate the power for such a three-phase system, the following equation is used:

$$P_{3\ phase} = \sqrt{3}V_{line-to-line} \times I_{line} \times PF \tag{39}$$

where $V_{line-to-line}$ refers to the line-to-line voltages and I_{line} refers to the line current. Any single line voltage and current can be selected for the calculation. PF refers to the power factor. CMTI's motor control software provider informed us that we can assume that the power factor is unity, due to on-board controllers balancing the load.

With movement and power sensors calibrated, the model calibration steps may be outlined.

3.2 MODEL CALIBRATION

The proposed digital twin comprises five models:

- Pose estimation model
- Force estimation model
- Traction estimation model
- Slip prediction model
- Terrain change detection model

Currently, the first two items are most sensibly classified as “models”, whereas the latter three items are better described as “algorithms”. Therefore, the current sub-section will only focus on the pose and force estimation elements, while the following sub-section will focus on the algorithms.

The pose estimation model (the Kalman filter) has already been discussed in detail in Section 2.2.1, and does not require further calibration. The parameters for the Kalman filter were determined by visual inspection; therefore, there is not much to report on this front.

The force estimation model relied on accurate platform mass and centre of mass data to correctly deconstruct the seven mass blocks outlined in Section 2.3.1. The center of mass location is also necessary to determine the ICR locations, which are measured relative to the platform COM. Furthermore, the platform mass is necessary to determine the power loss due to inclination. Therefore, careful experiments are required to capture these quantities as accurately as possible.

To determine the platform mass and centre of mass simultaneously, two lifting tests were conducted (see Figure 18). During the first lift, the platform must be lifted at the front using a load cell, and pivot about the rear wheels. During the second lift, the platform must be lifted at the rear using a load cell and pivot about the front wheels. Three distances must be measured before or after these lifts:

- The inter-pivotal distance (d) – the distance between the front and rear wheel.
- The pivot to front lifting distance (d_1) – distance from the rear wheel to the front lifting connection point.
- The pivot to rear lifting distance (d_2) – distance from the front wheel to the rear lifting connection point.

Furthermore, for the two lift exercises, the load cell force must be measured and noted during static equilibrium.The following assumptions are required to solve the static problem:

- The platform is rigid, and mass distribution does not change between lifts.
- The force measurements (F_1 and F_2) correspond to the forces required to lift the platform while it pivots about the wheels.
- The platform is in static equilibrium during each lift (no acceleration/dynamic effects).
- The platform is barely lifted off the ground, so the inter-pivotal distance is roughly d.

Based on these assumptions, the sum of moments around the pivot points for both lifting exercises is zero:

$$\sum M_{P1} = d_1 F_1 - x_{COM} mg = 0$$
$$\sum M_{P2} = (d - x_{com}) mg - \ d_2 F_2 = 0$$

where F_1 and F_2 represent the measured steady-state lifting force for lifting run1 and 2 respectively. The unknown mass and centre of mass are denoted by m and x_{COM}, respectively.

Two equations with two unknowns results, and can be solved for the mass and COM, resulting in the equations:

$$x_{COM} = \frac{F_1 \ \cdot d_1 \cdot d}{F_1 \cdot d_1 + F_2 \cdot d_2}$$
$$m = \frac{F_1 \cdot d_1}{g \cdot x_{COM}}$$

It is important to note that x_{COM} is measured as the distance to the COM measured from the rear wheel in the forward direction.

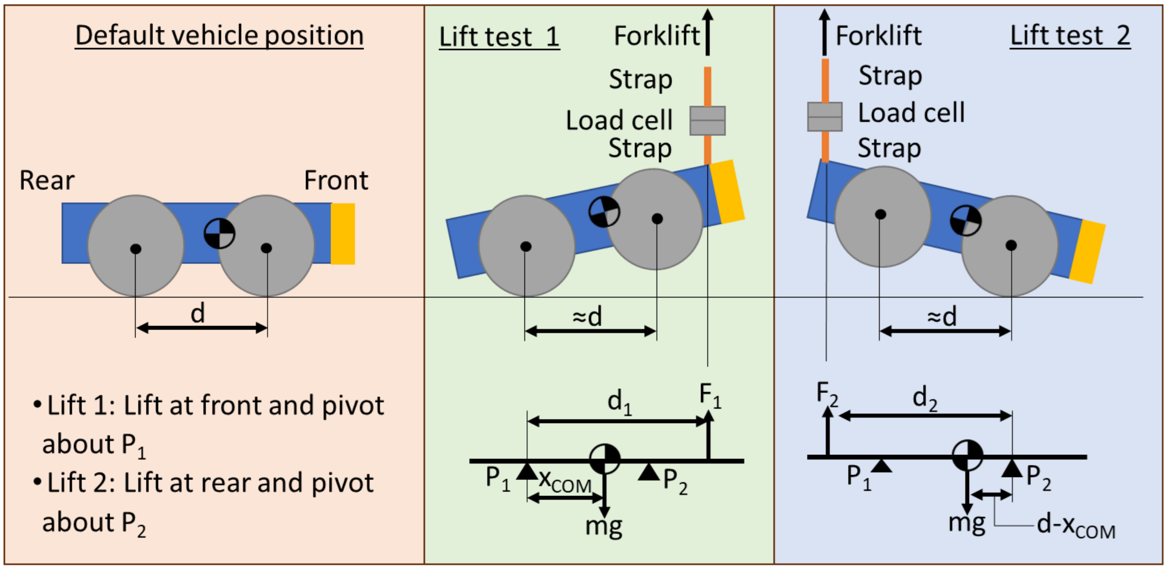


Figure 18 Planned lifting experiment to determine mass and centre of mass of the platform.

Table 7 summarises the values measured during the lifting tests and the calculated mass and COM.

Table 7 Results for the surface lifting tests.

Parameter	Measured value
d	1.38m
d_1	2.28m
d_2	2.65m
F_1	22500N
F_2	4200N
x_{COM}	1.13m
m	4612kg

From these results, the platform weight was calculated as 4612kg and had a centre of mass location 1.13 metres forward from the rear wheel. Observe that this is roughly 35cm behind the front wheel, which means the platform COM is severely biased forward by a large amount. This implies that the rear wheels will be much more likely to slip compared to the front wheels. This also implies sub-optimal torque distribution due to the front wheels needing to work harder when the rear wheels skid.

3.3 ALGORITHM CALIBRATION

As mentioned in the previous section, the algorithm calibration sub-section pertains to the traction estimation, slip prediction and terrain change detection algorithm. Each algorithm has unique considerations, which are discussed in their respective sub-sections below.

3.3.1 TRACTION ESTIMATION ALGORITHM CALIBRATION

The initial implementation of the traction estimation algorithm showed some signs of numerical instabilities, which prompted the need to calibrate the input data to the algorithm. The following three filtering measures had a large impact on stabilising traction results:

1. Implementation of a minimum time window.
2. Removal of transient conditions.
3. Removal of bad ICR values.

Additionally, the selection of optimisation algorithm parameters was unclear and, therefore, required calibration. Finally, surface experiments were conducted to estimate the platform's true longitudinal traction on surface, and needed to be calibrated. The following sub-sections will deal with each of these issues.

3.3.1.1 Input data filtering

The first filtering decision was to implement a minimum time window. Recall that the data is segmented according to periods of similar movement types. However, there are periods where the platform briefly enters one movement type before switching to another. This leads to extremely short optimisation segments and is likely not representative of the overall platform motion. Therefore, a minimum time window threshold is implemented. If a segment is shorter than this threshold, it is not considered for optimisation, and the data is ignored. This threshold was seen to have a significantly positive impact on the final results and is highly recommended. Low threshold values will lead to more regular updating but will likely be more sporadic because shorter, less representative samples are being evaluated. Higher threshold values will lead to more sparse updating events, but one can be more confident that these are likely more accurate and consistent.

A 5s threshold provided a good balance between update frequency and accuracy for this specific use case. However, this is likely dependent on the asset being used. The CMTI platform is designed to travel at low velocity and high torque, leading to slower ramp-up velocity times. This implies that a longer threshold period is specified to ensure steady-state behaviour is captured. A smaller time threshold would suffice for lighter, more nimble assets.

When looking at the movement segments, the first portion always contained some transient behaviour attributed to a ramp-up in platform speed. This is expected, as the platform is starting a new movement from being stationary. The current algorithm does not account for the power required to accelerate the platform up to speed, and therefore, these transients will negatively influence the final results. The first 10% of each selected window is removed to address this challenge, hopefully removing most transient, acceleration-based behaviour. The transient removal process significantly positively impacted the results and is highly recommended for large mining assets with steady-state algorithmic assumptions. Previous works have not addressed the transient speed challenge, likely because they have been applied to robot-scale vehicles with rapid transients that would not significantly affect the final result. It is important to evaluate algorithmic assumptions when implementing existing research, especially if developed for robot-scale applications but applied to large mining assets.

The third and final data filtering consideration is the removal of bad ICR locations, or more specifically, bad lateral ICR locations. Refer to the lateral ICR formulas:

$$y_{ICR,l} = \frac{v_{x,l} - v_x}{\omega_z} \tag{40}$$

$$y_{ICR,r} = \frac{v_{x,r} - v_x}{\omega_z} \tag{41}$$

When the platform is travelling in a straight line, ω_z tends to zero and the ICR value becomes very large. These points must be removed from the remaining data window, as these high ICR values will greatly affect the final estimate. If the data were perfectly clean, this would not be an issue, but because some noise will always be present, small noise effects at low speeds could have massive effects on the ICR values. Any ICR values larger than 100m were excluded to ensure the results remained bounded and not over-sensitive. Therefore, the final time window includes no turning radii larger than 100m. This is quite conservative, and this value could likely be lowered. However, at 100m, the results were quite stable. In fact, at 1000m, the results were also somewhat stable. Therefore, 100m is considered an arbitrary but sensible upper ICR threshold.

The ICR values have another associated challenge, which becomes clear when referring to the slip track:

$$B_s = y_{ICR_l} - y_{ICR_r} = \frac{v_{x,r} - v_{x,l}}{\omega_z} \tag{42}$$

If the platform is turning left, ω_z will be positive according to the global coordinate system. In this case, the left track velocity is smaller than the right track velocity. Two positive numbers are being divided, implying a positive slip track value. If the platform is turning right, ω_z will be negative according to the global coordinate system. In this case, the left track velocity is larger than the right track velocity. Two negative numbers are being divided, again implying a positive slip track value. Therefore, the slip track must always be positive irrespective of turning direction.

When evaluating the data, a few periods of negative slip track were found. One of three factors is likely to cause this issue. Firstly, if the sensor data is not perfectly aligned, physical contradictions may exist between the movement recorded from the CAN and LIDAR system. Data alignment should therefore be a high priority in the digital twin.

Secondly, some negative slip track values occurred near regions of low angular velocity. While the wheels try to turn the platform in one direction, the platform moves in the opposite direction. This should not be possible under high angular velocities. Still, one can expect that sensor noise would oscillate between positive and negative angular velocities during very low angular velocities, causing the slip track to oscillate accordingly. Depending on the severity of noise, this can be addressed by filtering, but some noise may always be present, and therefore pro-active measures need to be taken to exclude physically impossible data in the final twin.

Finally, one wheel set may have a lower traction than the other. For example, if the right wheel set was turning faster than the left side, yet the platform was turning clockwise, it could indicate that the slower-moving left track has much higher traction than the slipping, high-speed right track. This is quite an unlikely scenario, but it should not be discredited.

A minimum slip track threshold of 0.001m was assigned to counter these possible scenarios. One could be more conservative, perhaps choosing a 1m minimum value, but 0.001m worked well. This means that any anomalous slip track values are filtered from the dataset.

3.3.1.2 Algorithm initialisation

With the data completely filtered and ready for algorithmic implementation, three choices remained: Initial parameters, loss function and optimiser. In all three cases, we applied trial and error and came to the following conclusions: The algorithms didn't' appear to be sensitive to initial parameters, and therefore, we arbitrarily set initial traction values to 0.1 and the initial motion resistance coefficient to 1.

The complexity of parameter constraints also needs to be addressed. The motion resistance parameter is expected to vary with increasing resistance, and therefore this parameter is allowed to vary between 0 and positive infinity. For μ and μ_x, we expect values near or below unity but allow these parameters to range between 0 to 5 to account for high longitudinal traction values. The μ_y parameter is expected to be larger than μ or μ_x, as it measures the resistance to turning the platform. Skid steering movements are known to be power intensive, and we expect this parameter to be quite high during certain periods. We therefore allow this parameter to range between 0 and 20.

The loss function was tested between mean squared error and mean absolute error. This slightly impacted the final results, but not enough to justify a full sensitivity study. The mean squared error more harshly penalises outliers, whereas the mean absolute error penalises all data points equally. We used the mean squared error because Kalman filtering should have removed any noise-based outliers. Therefore, any large outliers are likely due to the true dynamic behaviour of the platform and should be penalised accordingly.

At this stage, two main questions remain:

1. Which optimizer algorithm should we use?
2. Is the optimization process leading to sensible results?

To answer the first question, six of Python's built-in optimisers were to determine which one(s) fail the optimization task least often, and which ones achieve the lowest total loss value. The total loss value is calculated by summing the absolute difference between predicted power and measured power at each timestep. The smaller the value, the better the optimization algorithm is doing at estimating accurate parameters.

The six algorithms being evaluated are Nelder-Mead, Powell, L-BFGS-B, TNC, COBYLA and SLSQP. Other algorithms exist, but they cannot handle the parameter constraints as discussed above. The result of the evaluation is illustrated in Figure 19.

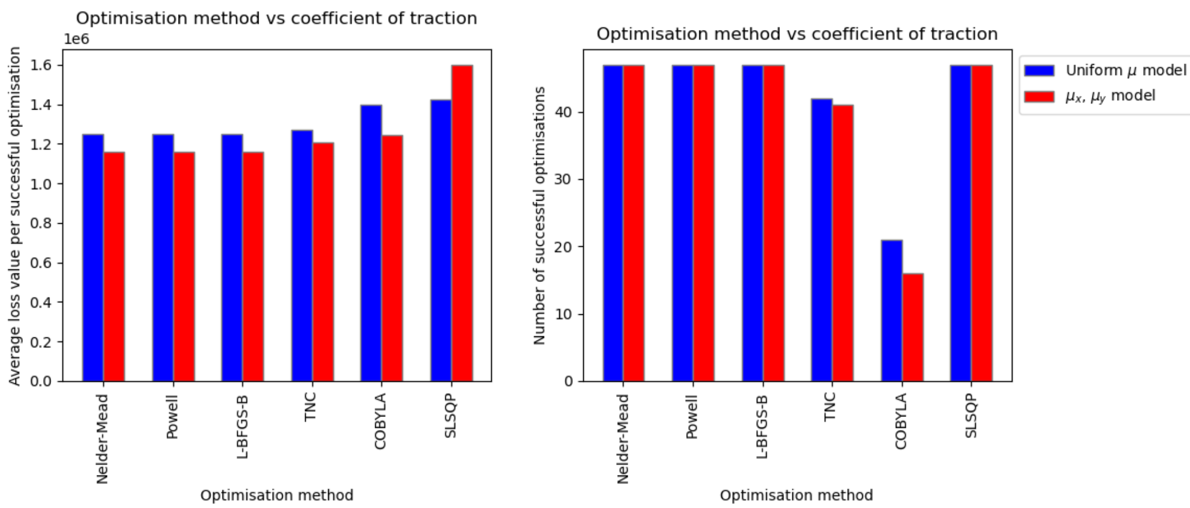


Figure 19 Evaluation results of the six optimisation algorithms that were considered.

From this evaluation, we can see that the TNC and COBYLA algorithms failed more often compared to the other algorithms, and are therefore eliminated from the pool of possible algorithms. Looking at the remaining algorithms, we see that SLSQP has a higher loss compared to the Nelder-Mead, Powell and L-BFGS-B algorithms and is therefore also eliminated from the pool of possible algorithms. The remaining three algorithms have identical performance, which likely indicates that we have set up the optimisation problem rigorously, so that three different optimisers converge on the same results independently. Also note that in every single case (except for SLSQP), the uniform μ model had higher loss values when compared to the novel proposed anisotropic traction model. This indicates that the proposed novel anisotropic traction estimation algorithm is a better traction model when compared to the isotropic model, because the anisotropic model can better fit the data, leading to lower loss values.

To answer the second question, we multiplied the measured power values by a scaling factor between 1 and 2, expecting the traction values to increase with increasing scaling factors. This is because a larger measured power implies that the terrain is resisting turning more harshly, and should therefore result in higher traction values. Therefore, we ran the traction estimation algorithm using the three remaining optimization algorithms for various scaling factors, and calculated the average traction values over the full run. The results are presented in Figure 20.

The results show a clear relationship where an increase in power leads to an increase in average traction. The L-BFGS-B algorithm deviated from the Nelder-Mead and Powell algorithms, and can therefore be removed from the pool of optimisers. However, looking at Nelder-Mead and Powell results, we see that they are indistinguishable, laying on-top of one another. These algorithms' results also show a monotonic increase, which is expected for this experiment. Finally, we notice that the mean μ_x value is always lower than μ, which in turn is always lower than μ_y. This is expected because the longitudinal traction is expected to be lower than the lateral traction. Furthermore, the uniform traction parameter needs to account for both directions' traction, and is therefore expected to be somewhere between the two extremes.

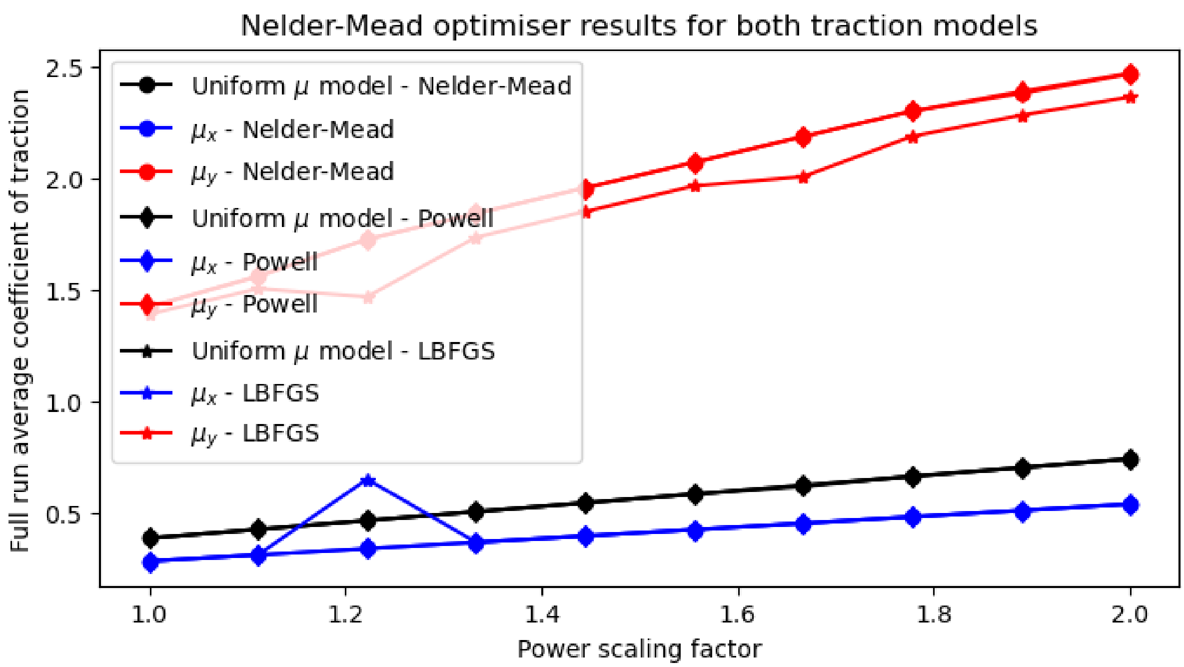


Figure 20 Comparison between average traction value and power scaling factor for various optimisation algorithms.

In conclusion, we have shown that the traction estimation results are consistent with expectations, and that either the Nelder-Mead or Powell algorithms may be used as robust optimisers for this task. We will arbitrarily use Nelder-Mead optimiser for the traction estimation algorithm going forward.

3.3.1.3 True surface traction

To fully validate the traction estimation model, some reference traction value needs to be measured on surface. An outline of the pull test is illustrated in Figure 21.

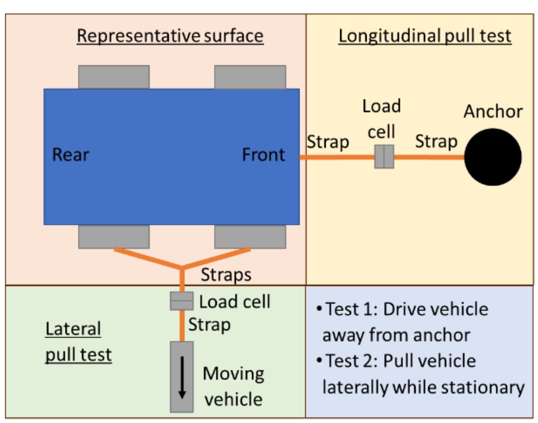


Figure 21 Planned longitudinal and lateral surface pull tests for the CMTI platform.

Test 1: Longitudinal pull test procedure:

1. Secure straps to the front of the platform, on either sides of the load cell and to an anchor point as shown in Figure 21.
2. Start data logging of calibrated load cell.
3. Slowly drive the platform in the opposite direction of the anchor until the straps are taut.

4. At this stage, the platform must pull with maximum torque in the opposite direction of the anchor until the wheels slip.
5. Continue pulling with the platform while wheels slip for about 3s.
6. Stop pulling.
7. Release tension in the staps.
8. Repeat steps 3-7, twice.
9. Finish data logging.

Test 2: Lateral pull test: Unfortunately, we could not arrange a lateral pull test due to severe time constraints and could only conduct a longitudinal pull test.

Using the data from the pull test (see Figure 22), we were able to estimate the static and dynamic longitudinal traction. To determine these values, we designed a model that accounted for the fact that the rear wheels would start slipping before the front wheels and that the pull test would induce a torque on the platform, leading to increased traction at the rear wheels.

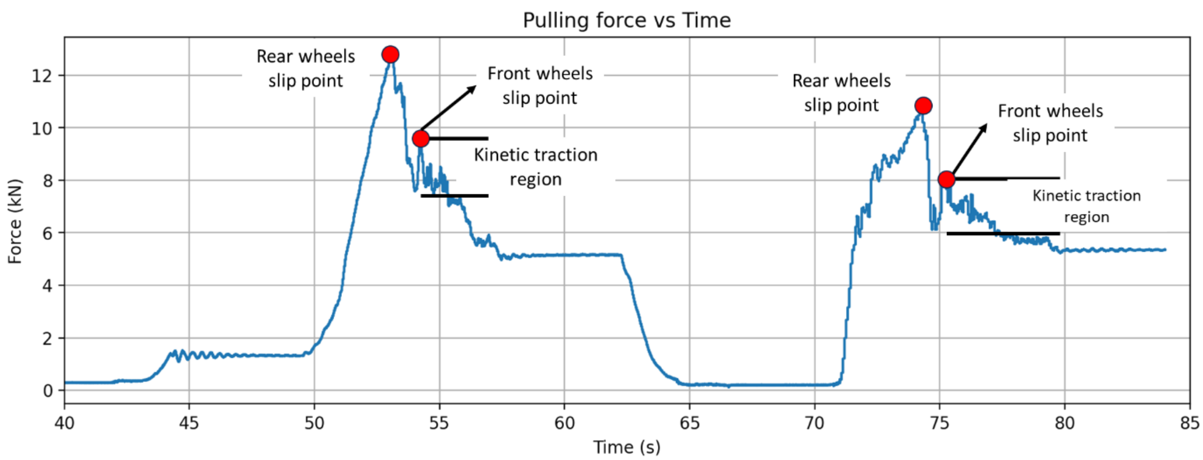


Figure 22 Analysis of pull test results. We see three major events for each test, namely a point where the rear wheels slip, then front wheels slip and a region where all wheels are slipping.

By selecting the correct static and dynamic traction values, we were able to find a good fit to the measured data. Figure 23 illustrates this model. Using this model, the static and dynamic coefficients were seen to be $\mu_s = 0.29$ and $\mu_k = 0.17$, respectively. These two parameters allowed us to form a traction bound, which describes the minimum and maximum possible longitudinal traction values of the CMTI platform at the specific test area on the site.

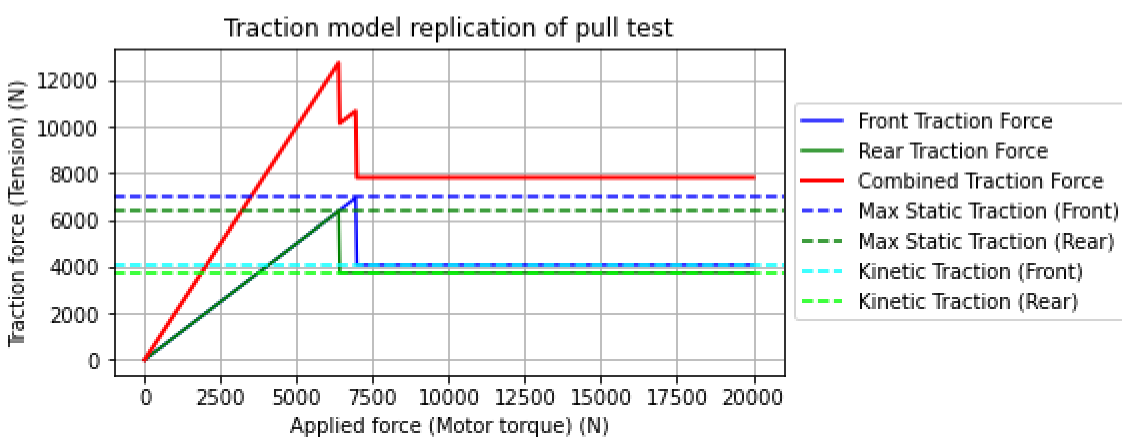


Figure 23 Traction model results under the assumption that all wheels experience unbalanced, non-steady state normal forces. Results are shown for $\mu_s = 0.29$, $\mu_k = 0.17$, $N_{front} = 24000N$ and $N_{rear} = 22000N$.

3.3.2 SLIP PREDICTION ALGORITHM CALIBRATION

To evaluate the success of the proposed approach, we need to detect wheel slip of the platform. For straight line movements, we need to find the conditions where:

$$r_i\omega_i - v_x > threshold \tag{43}$$

where v_x, r_i and ω_i represent the platform longitudinal velocity and the i^{th} wheel's radius and angular rotation rate. These quantities will never be in perfect balance due to sensor noise, mud caking up the wheels (causing a radius increase), and deflection of the tracks. Therefore, we need to incorporate some threshold of slip. For this project, we found a threshold of 0.07m/s to be sufficient. We increased the threshold value and compared the locations where slip was predicted against video footage to determine the threshold. Threshold values near 0.7m/s reliably predicted slip events that were visible on video footage.

3.3.3 TERRAIN CHANGE DETECTION ALGORITHM CALIBRATION

Currently the SVM's operation has been outlined, without addressing the calibration of the various algorithm parameters. The SVM has multiple parameters and implementation strategies. Initial testing revealed the following set of algorithmic settings to work well:

1. $\nu = 0.001$
2. $\gamma = 0.02$
3. The training data should use at least 5 minutes of data, of which the first 100s should be removed to allow all data acquisition systems to start logging.
4. The anisotropic traction and motion resistance model parameters should be used as model inputs.
5. A batch size of 30s should be used for evaluation.
6. A batch threshold of 95% should be used to identify "different" conditions.

These ν and γ parameters affect the boundary drawn by the SVM model. The selected parameters are not necessarily optimal, but they were found to be "good enough" for a first attempt at terrain classification. The third condition was necessary for the current project, because it often took about a minute for all data acquisition systems to be started prior to testing. Naturally, in a production environment, data acquisition happens continuously and therefore, this constraint is likely unnecessary.

The fourth condition specified the use of the anisotropic model. From initial testing, the anisotropic model showed a 10% improvement in correctly classifying similar terrain as being similar, and roughly a 13% improvement in identification of a change in terrain conditions. Therefore, the isotropic model is not considered for the digital twin.

The fifth and sixth conditions were determined from a parameter optimization study, and were shown to strike a good balance between terrain change estimate update frequency, true positive and true negative rates. Essentially, the SVM should evaluate a batch of points over a 30s period, and classify each data point as an inlier or outlier. If the proportion of outliers in that batch exceeds 95%, the terrain may be assumed to have significantly changed and the operator may want to take action.

4 DIGITAL TWIN IMPLEMENTATION (CASE STUDY)

Two sets of experiments were conducted at CMTI's surface test site to evaluate the digital twin's performance. The test site is on the same premises as the CMTI workshop and offices at the extreme end of the property. The drivable portion of the test site consists of soil. A photo and schematic rendering of the site are illustrated in Figure 24 and Figure 25, respectively.

Figure 24 Top) Coarse 3D rendering of CMTI's test site. This render was generated from LiDAR scans that were accumulated during experimentation. Drivable portions are shown in orange with a dark outline. Notice the ramp area at the top of the image. Bottom) Real view of the CMTI surface test site.

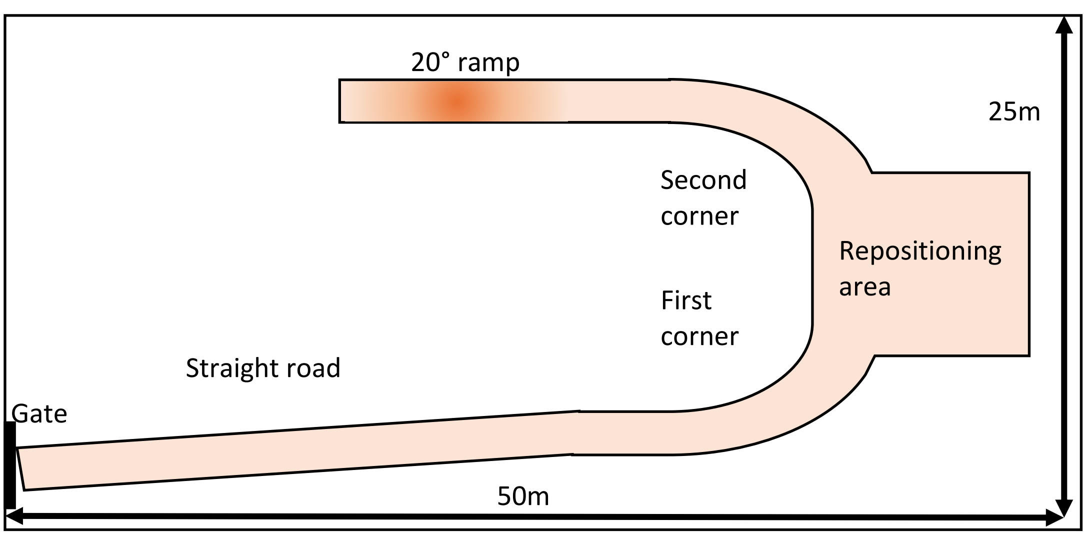


Figure 25 Top view schematic of the CMTI surface test site.

The test site is contained within an area roughly 50m by 25m. The portion from the gate is mostly straight and then bends left into the repositioning area. This area is used to change platform directions if necessary. When exiting this area using the second corner, a ramp is found where CMTI can investigate platform performance under high inclinations. The ramp becomes progressively steeper, with a peak slope of roughly 20°.

Typical commissioning runs will have the platform start at the gate, facing forward towards the first corner. The platform will then perform combinations of straight line driving and progressive steer to reach the repositioning area. Once here, the platform continues to move into position around the second corner to drive up the ramp in a straight line. Often, slight corrections in heading are necessary, and progressive or skid steer is attempted. After driving up the ramp, the platform slowly moves down the ramp again, in reverse. At this stage, the operator usually keeps reversing straight into the repositioning area, where the platform undergoes combinations of skid and progressive steer to get it pointing forward towards the first corner. The platform mostly uses progressive steer to turn through the first corner, and drives straight to the gate, facing the opposite direction to when it started. The test facility is, therefore, suited to test all the driving combinations of the CMTI platform, including straight line driving, progressive steering, and skid steering.

The surface experiments could only be conducted once the CMTI platform had been constructed and commissioned. Unfortunately, unforeseen delays caused the commissioning date and the date the platform would be sent to the mine site to converge. Therefore, we were not granted a dedicated day of testing, but would rather be allowed to test during planned final commissioning runs before the machine was sent off to the mine. This restricted the available time to complete experiments and therefore a single run was captured on the planned day of experimentation, 17 October 2024. The mine delivery was delayed and we were allowed to conduct another run during a second commissioning run on the 19th of October. Both runs were fairly long (exceeding 10 minutes) and were comprehensive enough to evaluate the traction, slip prediction and terrain change algorithms.

Both days' testing followed the route and order outlined in Figure 26. The platform started on the ramp (1) and moved down towards the repositioning area. To get into the repositioning area, strong skid or progressive movements were made (2 and 3). Once in the repositioning area, the platform performed various movements to point towards the gate (4). Once correctly oriented, the platform steered around the first corner using progressive steering (5). Once aligned, the platform could proceed in relatively straight movements toward the gate (6). On the 17th of October, the platform reversed (7) after movement 6, and then went forward again (6), leading to the final movement, which required the platform to skid steer to reposition itself (8). The repositioning was necessary to ensure the platform could drive through the narrow gate.

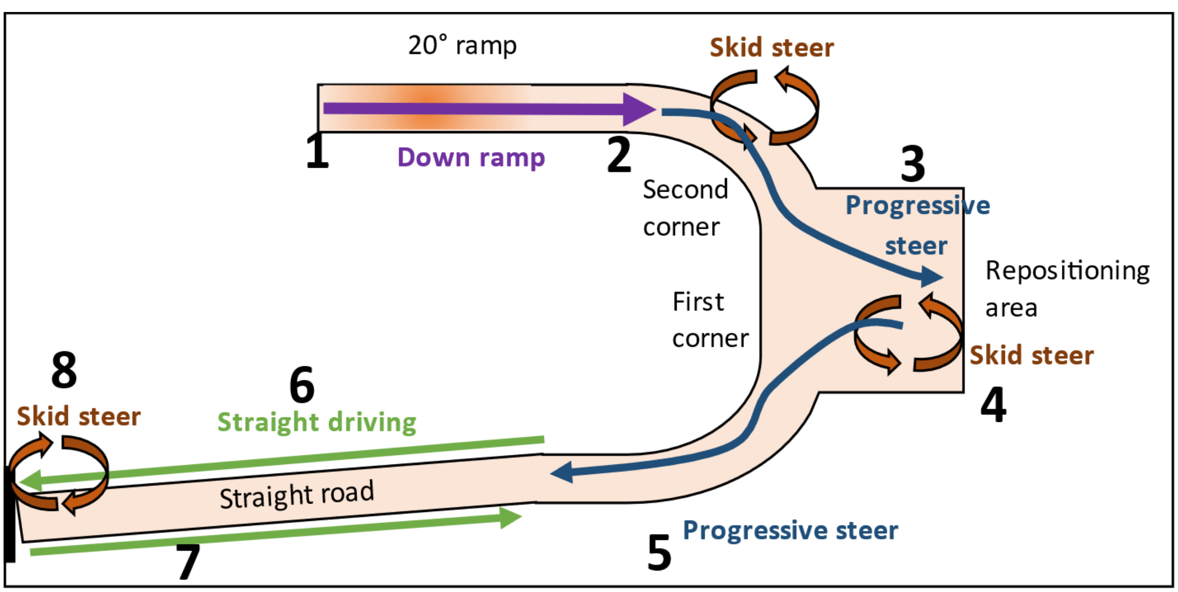


Figure 26 General route followed by the CMTI platform for both surface tests.

On the 17th of October the platform started pointing down the ramp, whereas on the 19th of October the platform started pointing up the ramp. This difference in initial orientation implies that the platform movements were not exactly as outlined in Figure 26, but were roughly equivalent. The platform traversed similar terrain areas, which will be shown later.

The surface conditions between the 17th and 19th of October differed due to the weather leading up to the various experiments. During the night of the 15th of October, Midrand (the area of the test site) received a large amount of rain. The following day had cool weather, meaning the rain did not have time to dry up for surface experiments on the 17th of October. Consequently, the test site was dry on the surface but quite muddy as the platform dug into the surface during various movements.

Following the 17 October experiments, warm conditions on the 18th October and morning of the 19th of October allowed the test site to dry up a bit more. Consequently, when the platform attempted various movements, the conditions beneath the surface layer were much dryer.

The impact of the weather conditions is illustrated in Figure 27. In the first comparison, we highlight the bricks to the left and the same rock to show that we are looking at the same test site area, 2 days apart. We observe that the 17th of October has more damp conditions, and a caked up right rear wheel. The 19th of October still shows some moisture in the deeper soil layers, but for the most part the soil was dry, with almost no caking on the rear right wheel.

Figure 27 Comparison between the 17 October and 19 October surface runs, illustrating the dryer conditions on the 19th of October.

For the second comparison, we highlight the glove to indicate we are comparing similar areas of the terrain. On the 17th of October we observe quite muddy soil, which has just been kicked up by a platform movement. Consequently, the rear right wheel is also quite caked up. The 19th of October shows dryer conditions, with a much less caked rear right wheel.

These comparisons show that the surface conditions at CMTI were demonstrably different between the two days of running. Therefore, we should treat the two days' experiments as "different" sites instead of combining both days' data into a single dataset. This is advantageous, because it allows us to compare the traction estimation algorithm between the two days, to see if a statistically significant difference in traction estimates is found between the two days of running.

Finally, it is important to mention that in an attempt to shift the centre of mass away from the front wheels, CMTI welded 150kg of plates to the rear of the platform after the 17 October tests, but prior to the 19 October tests. The model calibration steps were repeated as outlined before, ensuring the collected data remains correct for the 19th of October.

The following sub-sections will outline various analyses to evaluate each traction estimation, slip prediction and terrain change detection algorithm. The results will be shown and analysed for each investigation.

4.1 TRACTION ESTIMATION ANALYSES

The following sub-section will evaluate various traction algorithm parameters across two main variables: testing date and model type. The 17th of October test conditions are considered more "muddy" and will be analysed accordingly. The model type refers to the isotropic or anisotropic traction formulation. Recall that the isotropic model assumes a single traction value when modelling slip power losses, whereas the anisotropic model splits the power loss contribution across longitudinal and lateral components.

The following set of analyses are deemed meaningful in exploring the efficacy of the traction estimation algorithm and will be reported in the following sub-sections:

1. Traction coefficient(s) analysis:
   a. Variation of traction values over time.
   b. Comparison of traction distributions across different testing days.
   c. Anisotropic traction coefficients' ratio over time.
   d. Comparison of anisotropic traction ratio distributions across different testing days.
   e. Traction coefficients' variation over x-y site map.
   f. Comparison of traction coefficient ratio between days on x-y site map.
2. Slip track analysis:
   a. Variation of the slip track over time.
   b. Comparison of slip track distributions across different testing days.
   c. Slip track variation over x-y site map.
3. Motion resistance analysis:
   a. Variation of the motion resistance over time.
   b. Comparison of motion resistance distributions across different testing days.
   c. Motion resistance variation over x-y site map
4. Power modelling analysis:
   a. Power modelling accuracy between isotropic and anisotropic models.
   b. Algorithmic loss comparison between isotropic and anisotropic model types.

Each of the following sub-sections outlines the analysis setup, results and interpretation of these results. Note that various plots will be shown with banded colours. These represent the movement of the platform. The legend is not shown, as it clutters the figure. Therefore, the following legend may be used:

- Gray – Neutral
- Green – Forward
- Yellow – Forward progressive right
- Dark purple – Skid right
- Light purple – Skid left
- Red – Reverse
- Pink – Reverse progressive left
- Light blue – Reverse progressive right

4.1.1 TRACTION COEFFICIENT(S) ANALYSIS

4.1.1.1 Variation of traction values over time

The traction values for both isotropic and anisotropic models were computed for the 17th and 19th of October, and are plotted over time as illustrated in Figure 28 and Figure 29, respectively.

For both days' testing, the isotropic traction (μ) is always sandwiched between the anisotropic model's results for μ_x and μ_y. Considering the traction ellipse concept, both the lateral and longitudinal traction coefficients must be identical to draw the equivalent isotropic circle. To draw an ellipse, one coefficient must be larger than the circle-equivalent parameter and the other smaller. Therefore, the obtained results are consistent with mathematical expectations. Note that the optimisation algorithm is run independently for the isotropic and anisotropic case. Therefore the sensible results showcase that the underlying algorithm is working correctly.

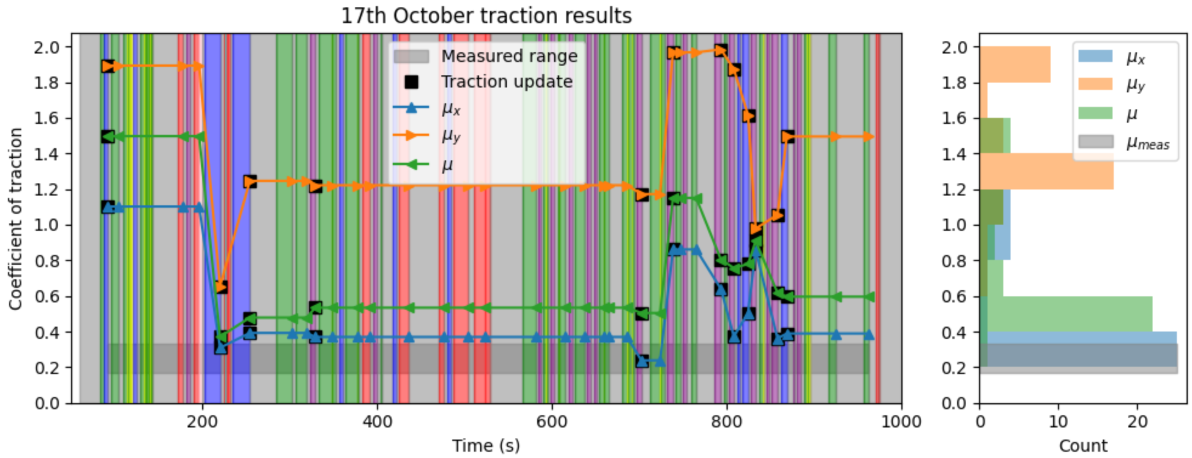


Figure 28 Traction values for the isotropic and anisotropic models over time for the 17th of October.

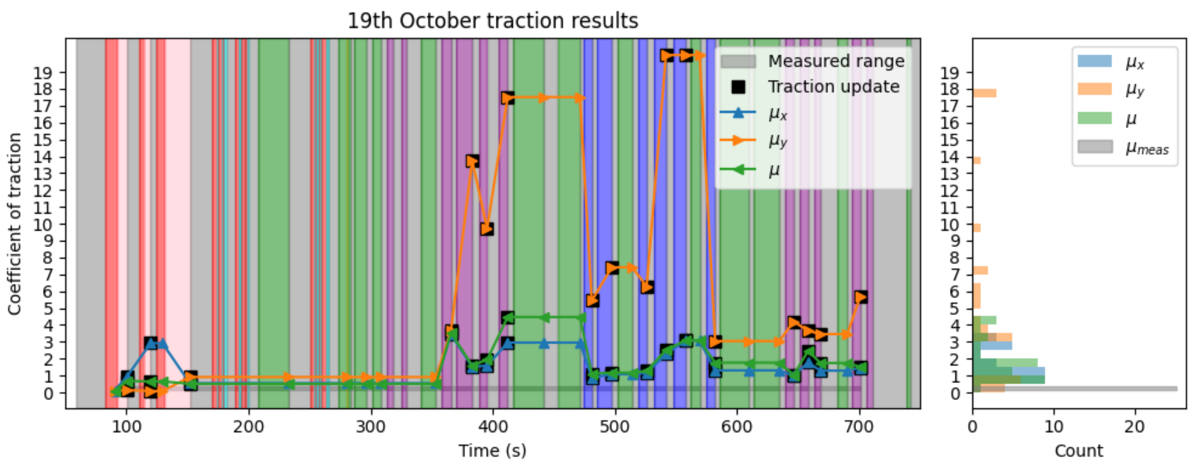


Figure 29 Traction values for the isotropic and anisotropic models over time for the 19th of October.

The results also showcase that $\mu_x < \mu_y$ for all points across both days (except near the start of the 19th of October). This need not be the case but is expected if the physics of the situation is considered. The longitudinal traction coefficient describes the resistance the wheel experiences in the longitudinal direction while turning. Considering the wheel digging into a muddy surface, one can appreciate that the longitudinal cross-section has much less area in contact with mud than a lateral cross-section. Therefore, the force required to overcome the sheer force of the mud is expected to be smaller for the longitudinal direction than for the lateral direction. Consequently, the traction coefficients reflect this expectation. However, the platform was reversing down the ramp at the start of the 19th of October's run. This means that the platform would have comparatively more traction due to the gravitational component. The opposite would be true if the platform moved forward up the incline. Therefore, the reversed relationship between μ_x and μ_y is likely due to the incline.

Finally, refer to the grey band that runs across both days. This band represents the measured longitudinal traction values of 0.17 to 0.33 outlined in Section 3.3.1.33.3.1. On the 17th of October platform reached the same area at about 700s into the run in Figure 28. At this point, the estimated longitudinal traction aligns with the measurements. It is important to reiterate that the traction algorithm estimates traction values independently from the pull test measurement, and the alignment between the algorithm and surface test is an independent validation of the algorithm. We cannot comment on the 19 October run because conditions were different and cannot be compared to the surface pull test on the 17th of October.

When considering the isotropic traction coefficient for the 17th of October, especially at 700s, one realises that the value is too high when compared to the surface pull test. This is expected because the isotropic traction coefficient simultaneously has to characterise both longitudinal and lateral traction effects. Since the lateral traction value is quite high at the 700s mark, the isotropic traction coefficient is expected to be "pulled upwards", away from the longitudinal value.

In summary, two points must be emphasized:

1. The traction model met all expectations, passing all sanity and independent experimental checks.
2. The anisotropic traction model shows clear areas where the anisotropic traction coefficients significantly differ from the isotropic traction coefficient. This indicates that the flexibility afforded to the proposed traction model is sensible and necessary to describe the true traction experienced by the wheels. An isotropic assumption is, therefore, insufficient and oversimplifies the relationship between the modelled and measured power.

4.1.1.2 Comparison of traction distributions across different testing days

To better visualize the degree of difference between traction parameters across testing days, Figure 30 is presented. The figure outlines the distribution of traction estimates across both days of testing.

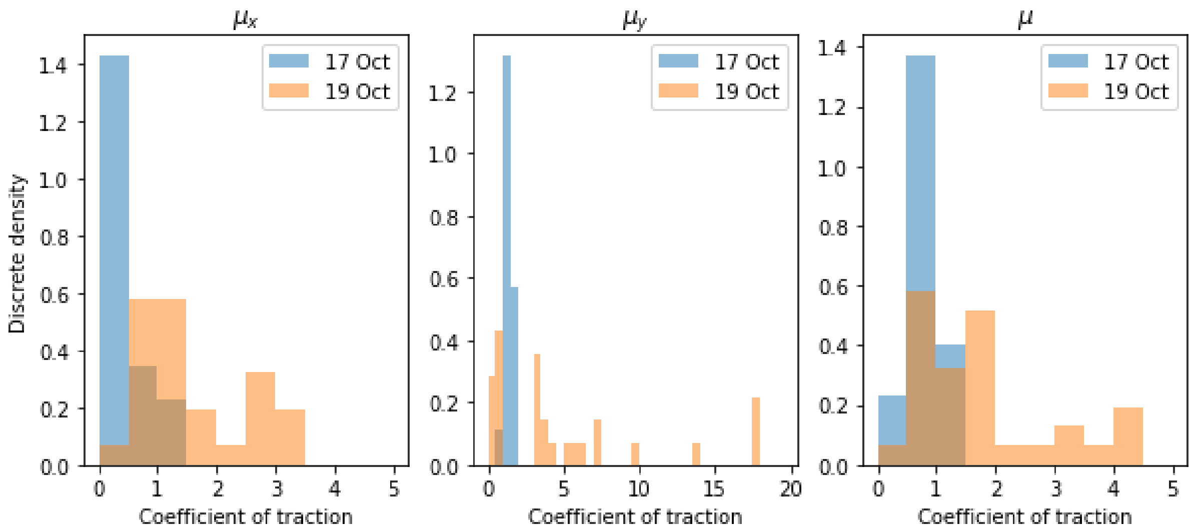


Figure 30 Traction distributions for the isotropic and anisotropic models across both days.

When comparing the results over both days, the terrain on the 19th of October had significantly higher traction coefficients. Table 8 summarises the mean traction values for both days:

Table 8 Mean traction coefficients over both days of testing for both model types.

Day	Isotropic traction (μ)	Anisotropic traction (μ_x)	Anisotropic traction (μ_y)
17 October	0.72	0.52	1.41
19 October	1.70	1.57	6.50

Notice that traction values are not necessarily smaller than 1, as required from a "friction" coefficient. This fact emphasizes that the algorithm is not measuring friction (which should theoretically be bounded between 0 and 1). The traction coefficients measure the friction between the wheel and the surface and other factors like the force required to shear through the soil. These additional resistive forces create situations where the platform must apply a greater force than the normal force to start skidding, leading to larger-than-unity traction values.

Looking at all traction coefficients, some overlap is seen between both days, but the 19th of October results show greater variance in traction values. This indicates that the terrain is less uniform on the 19th of October, leading to a large parameter variance. This finding aligns with surface conditions. On the 17th of October, the entire test site was quite muddy, leading to roughly similar terrain conditions. However, on the 19th of October, the surface was dry, with mud deeper down. Therefore, higher traction values were expected when the platform didn't break the surface layer. However, once the surface was broken, muddier conditions were experienced, leading to lower traction values.

4.1.1.3 Anisotropic traction coefficients' ratio over time

To truly appreciate the necessity for an anisotropic model, in Figure 31 we plot the traction ratio $\left(\frac{\mu_y}{\mu_x}\right)$ against time for both days' runs.

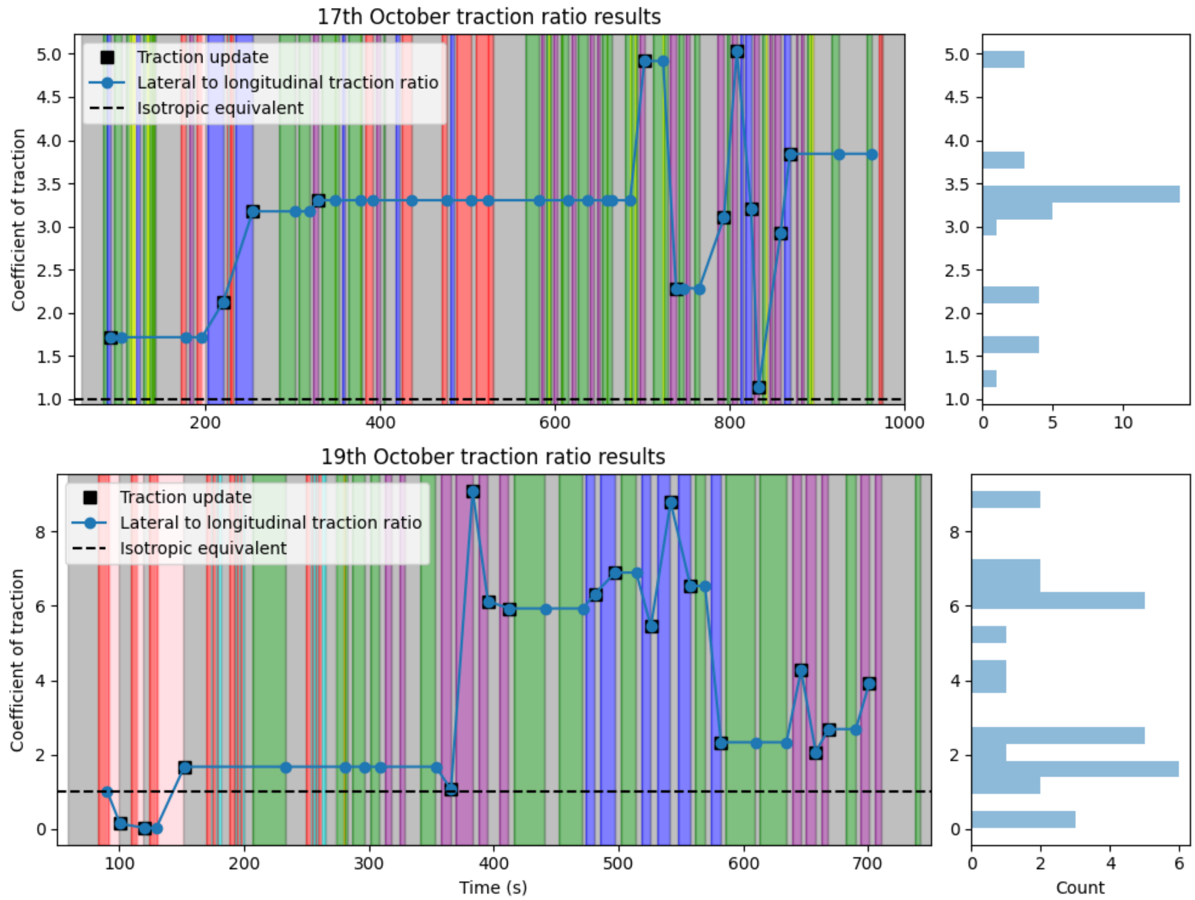


Figure 31 Ratio of longitudinal to lateral traction for the anisotropic model. 17 October's run is shown at the top, with the 19th of October's run shown at the bottom.

These results produce three important insights. Firstly, the ratio between the longitudinal and lateral traction is not constant, meaning the relative contribution of each parameter to the platform's power usage varies over the terrain. This further implies that the traction ellipse's shape constantly varies as the platform drives around.

Secondly, the ratio is almost always above unity, implying that the lateral movement of the platform has a much higher resistance than the longitudinal movement.

Finally, the ratio between the parameters is generally not near unity. This is important because unity represents the isotropic traction case, where $\mu_x = \mu_y$ and consequently $\frac{\mu_y}{\mu_x} = 1$. The results, therefore, emphasize that the turning behaviour of a skid-steer vehicle is vastly oversimplified when an isotropic assumption is made.

Considering all three findings, we may conclude that, on average, an isotropic model is insufficient to describe the underlying interaction between the platform wheels and the terrain. Furthermore, we can conclude that the lateral resistance is almost always higher than the longitudinal resistance and that the ratio of these two parameters constantly changes.

4.1.1.4 Comparison of traction ratio distributions across different testing days

The distribution of traction ratios across both days of testing is compared in Figure 32. For the 17th of October, traction ratios of 3 are likely, whereas results are much more varied for the 19th of October. The muddy conditions on the 17th of October lead to more consistent traction ratios, indicating a uniform test surface. In contrast, the dryer 19th of October test site leads to more varied traction ratios, indicating a varying test surface.

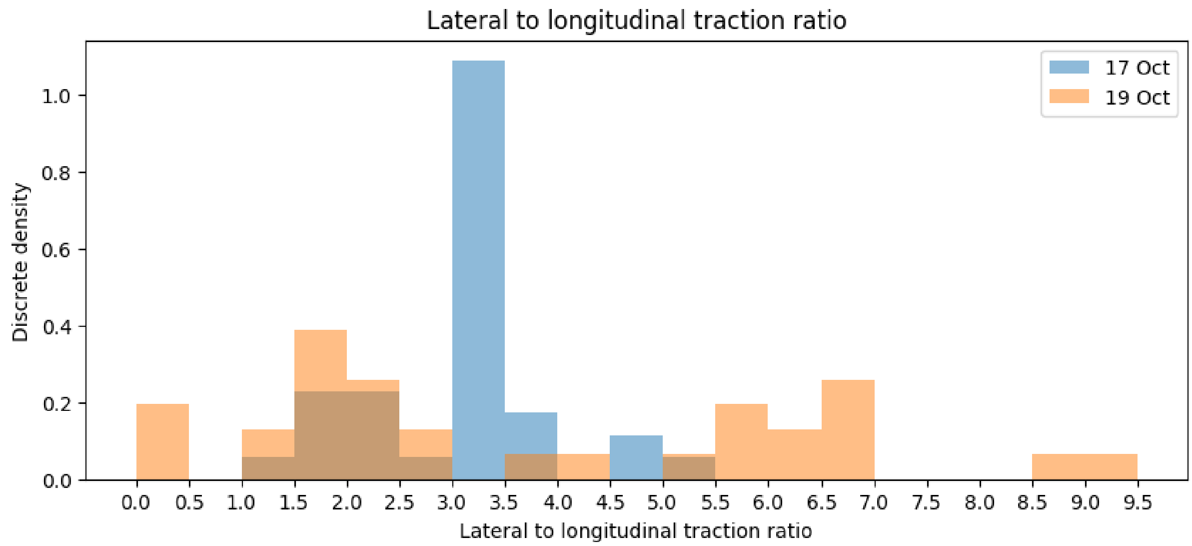


Figure 32 Traction ratio distributions across the 17th and 19th of October testing runs.

4.1.1.5 Traction coefficients' variation over x-y site map

Figure 33 illustrates the variation of the longitudinal and lateral traction coefficient values as the platform travelled over the terrain for both test days. The (0,0) coordinate for both days represents the ramp, and the final coordinates at the bottom left, the area near the gate of the test site.

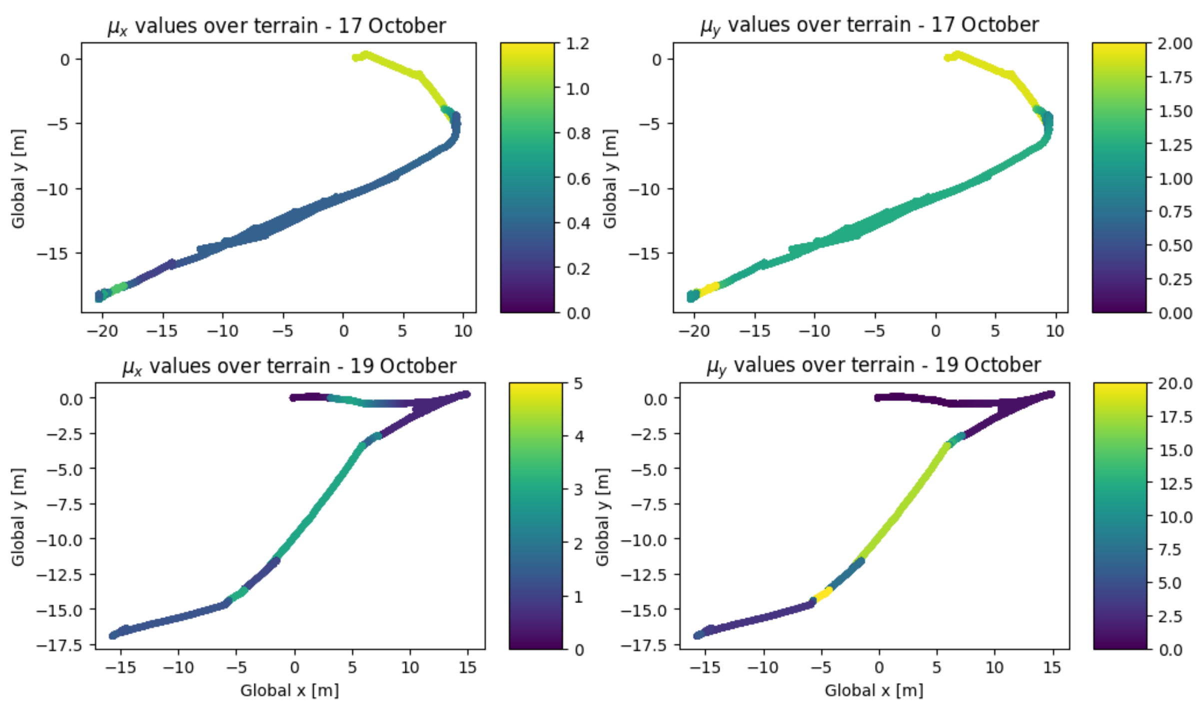


Figure 33 Visualisation of anisotropic traction values for the 17th (top) and 19th (bottom) of October test runs.

For the 17th of October, high traction values are found for both directions near the ramp area. Therefore, this area's surface was relatively dryer, and these values are expected. After going around the bend, both values decrease. This is also expected because the linear area of the plot had quite muddy conditions, and lower traction values are sensible. Finally, near the gate, traction values increase. The platform started digging into the mud, meaning movement was more restrained, leading to higher traction values.

For the 19th of October, longitudinal traction values were relatively high near the start of the run. Interestingly, in contrast to the 17th of October, the platform started by reversing down the incline, whereas on the 17th, the platform drove down the incline in a forward direction. The orientation of the platform had a mentionable effect on the traction values. Due to the platform reversing down the incline, the front-heavy platform could distribute more weight onto the back wheels, leading to greater mass distribution and better traction.

After reversing down the incline, the platform changed direction, going forward towards the gate. During this period, the traction values initially decrease but then increase significantly. Looking at reference video footage, the platform was on terrain previously undisturbed and dry during this portion of the run. This implies that shearing the soil during turning requires much power, and consequently, the traction values are high. Later in the run, where values start to decrease, the footage shows the platform travelling over previously disturbed, wetter soil, lowering the power required to shear the soil for turning. A final spike in traction is seen later in the run near the coordinates (-5, -15). Referencing the video footage, this occurred on a patch of soil with some grass, meaning the shear force required to turn would be quite large, increasing traction. After a short drive forward away from the grass, the platform skid quite easily, and consequently, the traction values were seen to decrease significantly.

When referencing the footage, during skid movements, sometimes one of the rear wheels slips, digging into the hard, dry surface, while the other is stationary. Then, when the platform tries to turn, the wheel that has dug itself into the surface, cannot overcome the soil's shear force. In these instances, the platform must drive forward/backwards for a short distance to remove the wheel from the dip, allowing it to reattempt a skid movement.

In conclusion, traction values are expected to increase when wheels sink into previously undisturbed, dry, soil-based surfaces. However, traction values are lower if travelling over previously disturbed, wetter soil. This implies that if a mining environment has a soil-like environment, areas where vehicles commonly travel will have lower traction than areas where few vehicles travel.

4.1.1.6 Comparison of traction coefficient ratios between days on x-y site map

Figure 34 illustrates the traction ratios for the 17th and 19th of October as a function of the terrain location. The bottom figure compares both runs on the same global map, allowing for comparisons.

For the 17th of October, the traction ratio increases as the platform moves towards the gate, and then decreases again when nearing the gate. On the 19th of October, a similar pattern emerges. When comparing the two runs on the same global map, good correspondence is found between the two runs. Near the ramp area, traction ratios are smaller. This indicates that the longitudinal and lateral resistances are roughly equal. This is likely due to the undisturbed terrain in this area, leading to easier skid movements.

When nearing the straight portion of the run, traction ratios increase, showcasing that lateral movements are becoming comparatively more difficult in relation to longitudinal movements. This can likely be attributed to the platform travelling over previously disturbed terrain, and the wheels have a larger chance of sinking into existing trodden terrain, causing easy longitudinal skidding but difficult lateral skidding. Finally, values increase further near the gate, which may be attributed to the platform sinking into the surface.

Considering these results, the traction ratio is a good indicator of terrain conditions. A terrain classification algorithm should, therefore, include both longitudinal and lateral traction, as the combination of both pieces of information is quite informative of terrain conditions.

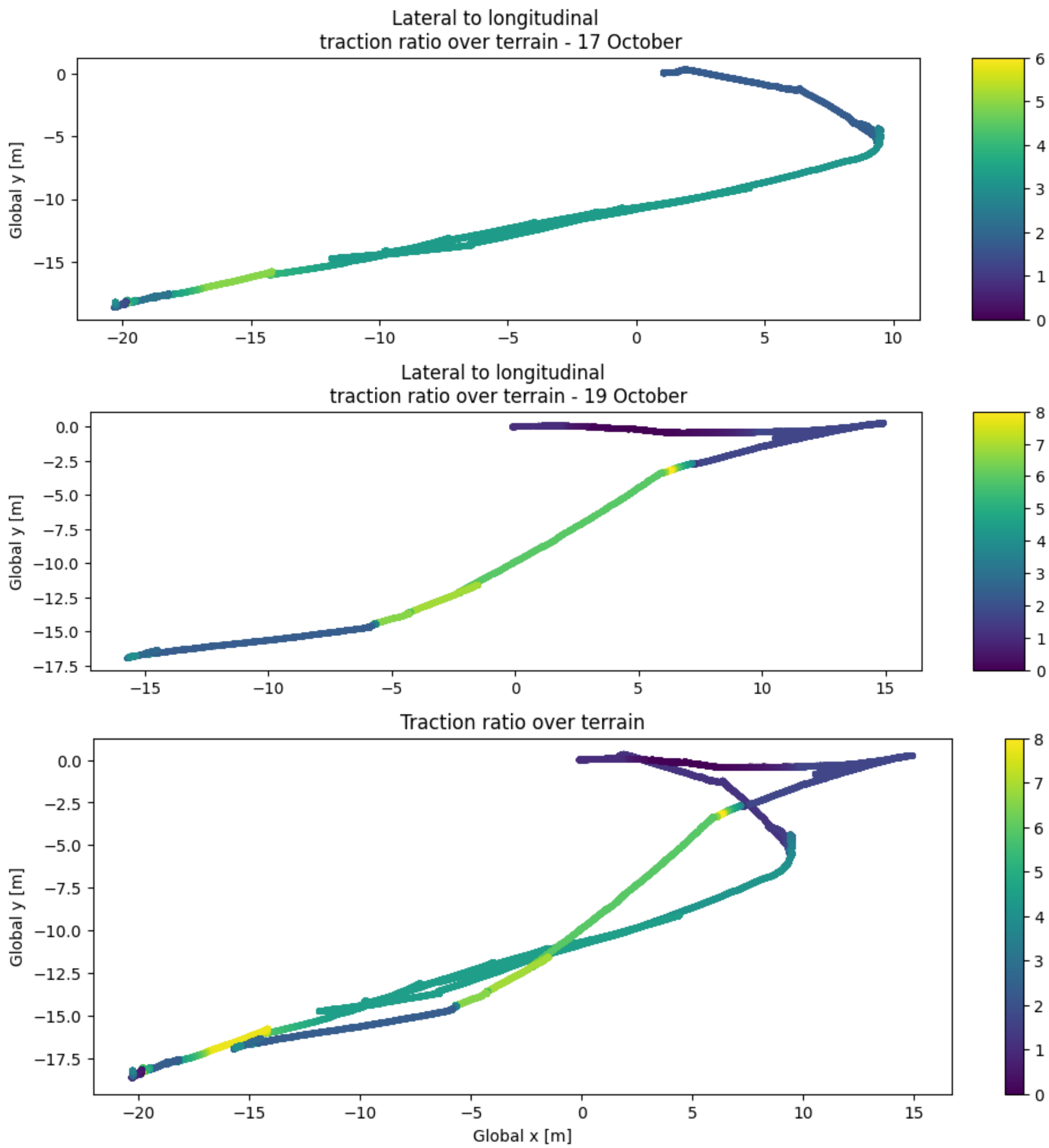


Figure 34 Variation in traction ratio values over the CMTI surface tests for the 17th and 19th of October.

4.1.2 SLIP TRACK (B_s) RESULTS

4.1.2.1 Variation of the slip track over time

The slip track values (B_s from Section 2.3.2) are less interpretable compared to the traction values, but do shed some insight into the terrain conditions and algorithmic stability. The slip track values are plotted over time in Figure 35.

Although the slip track values aren't equal for both days, they seem to have a similar unimodal distribution, with the mode being present at small slip track values. The fact that the 19th of October has larger slip track values, likely points towards the fact that the terrain had more traction, meaning slower rotation rates were found for the same input power, leading to larger slip track values, on average.

Slip track values are expected to be identical for both anisotropic and isotropic models because the slip track is completely independent of the traction formulation, only relying on wheel speed and turning speed, which do not require any traction information.

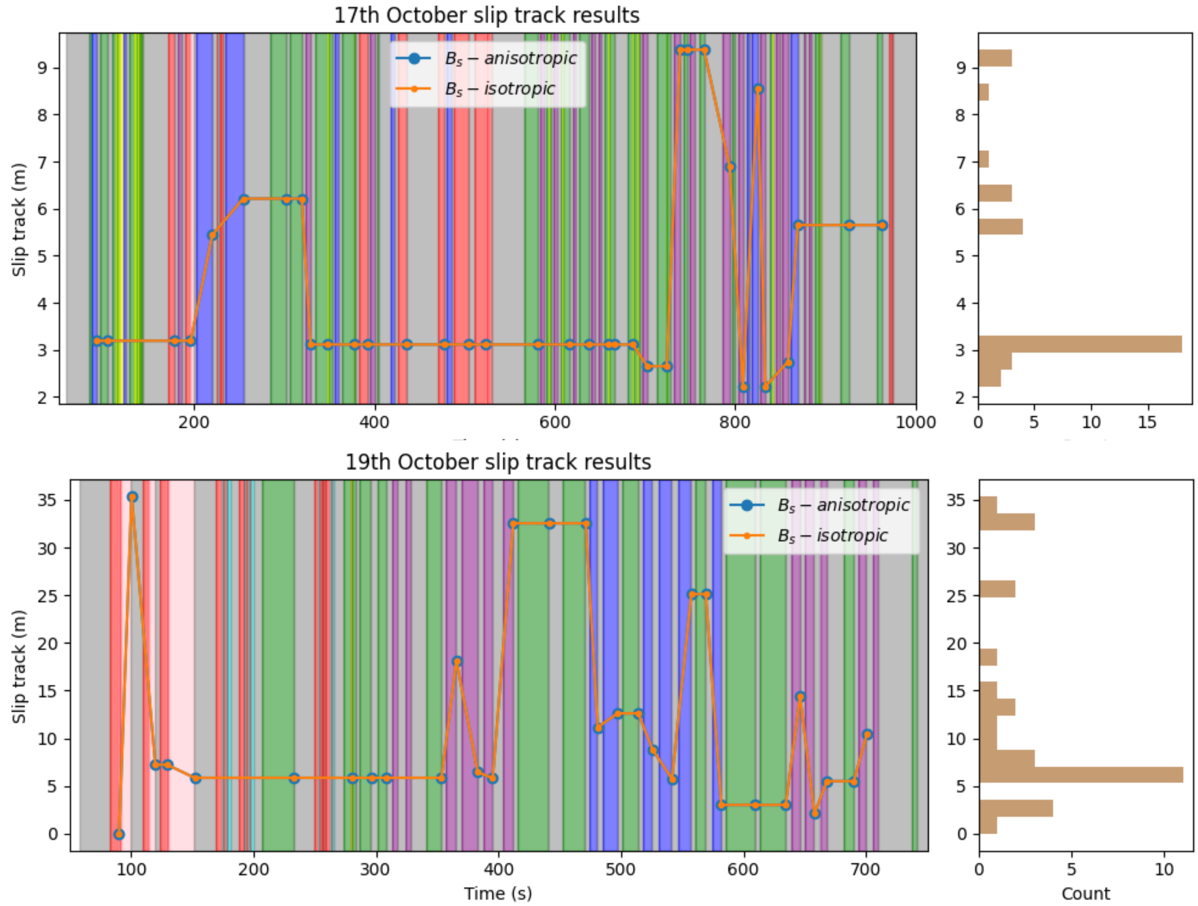


Figure 35 Slip track values for the isotropic and anisotropic models over time for the 17th (top) and 19th (bottom) of October.

4.1.2.2 Comparison of the slip track parameter across different testing days

Figure 36 represents the distribution of slip track values across both days of testing for both model types. The figure indicates a strong overlap between small slip track values for both days, with the 19th of October having a few larger slip track values, likely due to the slower platform rotation speeds for the same wheel velocity inputs.

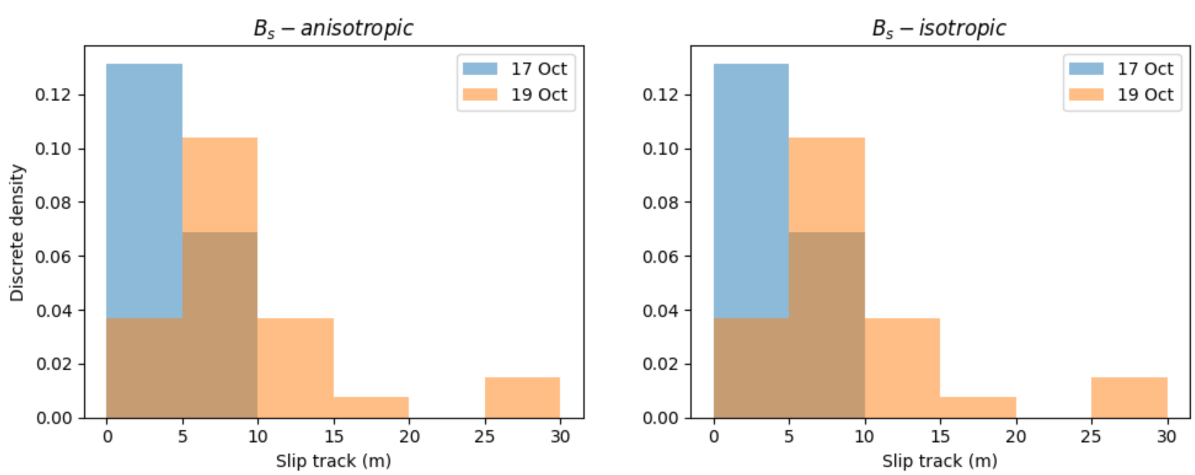


Figure 36 Slip track distribution comparison across both testing days, for both model types.

4.1.2.3 Slip track variation over x-y site map

In Figure 37 the slip track is plotted as a function of terrain location for both days. For the 17th of October, the highest slip track values are present when turning out of the ramp area, and when nearing the gate. Video footage showed that the turn, exiting the ramp area, was quite laboured and slow, likely leading to a higher slip track value. However, large slip track values near the gate are attributed to the fact that the platform dug itself into the surface, making turning very difficult/slow, leading to large slip track values.

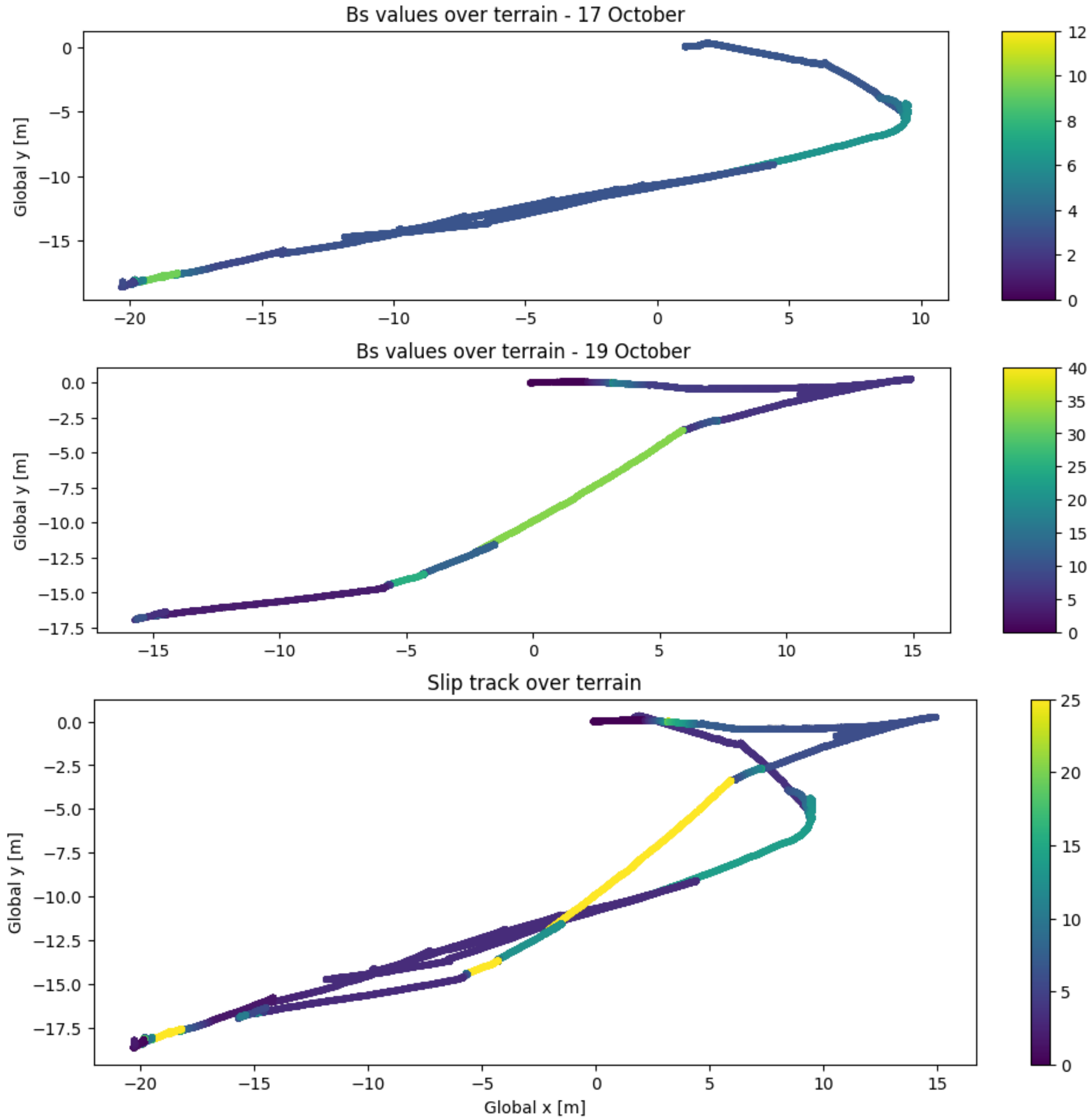


Figure 37 Slip track values over the global terrain. The top figure represents the 17th of October in isolation, the middle figure the 19th of October in isolation and the bottom figure both plots superimposed across the same global test site.

For the 19th of October, the largest slip track values were found between the ramp area and the gate. Looking at the footage, this large slip track value may be attributed to the platform's difficulty completing a turning movement, leading to a low rotation rate and high slip track.

Both trajectories are plotted on the same global map to simultaneously evaluate the slip track across both days. This map shows relatively low slip track values throughout, with some overlap between higher slip track values across the two days, especially near the gate. This may indicate that the slip track parameter is another useful parameter for terrain classification. However, the terrain change algorithm presented later will reveal whether this is true.

4.1.3 MOTION RESISTANCE (G) RESULTS

4.1.3.1 Variation of the motion resistance coefficient over time

Figure 38 depicts the motion resistance coefficients for both testing days. Note that the motion resistance coefficient is only updated during straight-line driving or progressive steering. For straight-line driving, only the motion resistance coefficient is updated, and therefore, this update is independent of model type (isotropic or anisotropic). However, motion resistance and traction parameters are updated during progressive steer movements. For the 17th of October, no progressive steering movements were sufficient for parameter updating and therefore we see identical results between the two model types. For the 19th of October, only two progressive steering movements were used, and therefore the two models' results are almost identical.

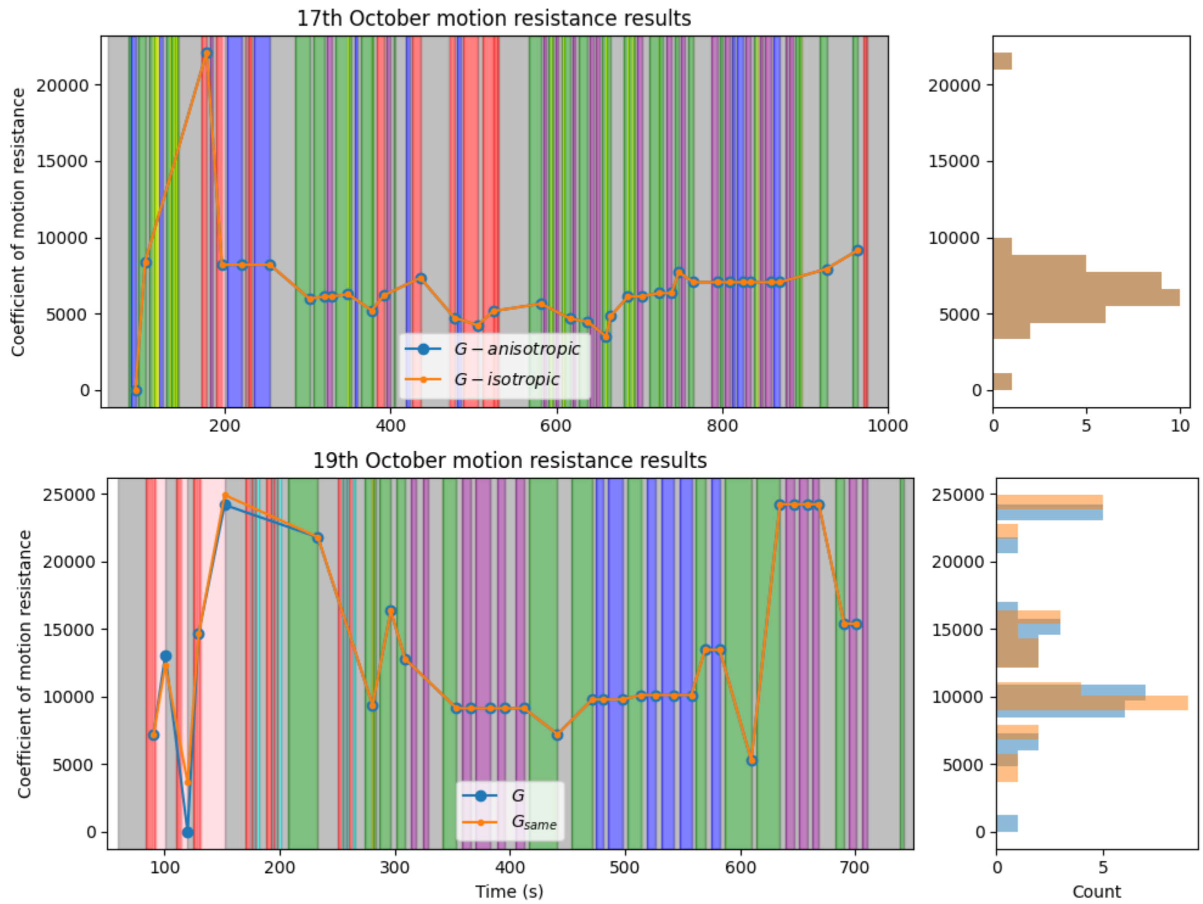


Figure 38 Motion resistance coefficients over time for the 17th and 19th of October.

For the 17th of October, motion resistance results are relatively stable, only increasing near the start and end of the runs. Similarly, on the 19th of October, motion resistance values were high near the start and end of the runs. This indicates that the terrain near the start and end of the run was likely more challenging, and therefore resulted in higher motion resistance values. Finally, the 19th of October showcased slightly higher average motion resistance values, likely due to the platform being 150kg heavier compared to the 17th of October, implying more power is being used to drive the platform at the same velocity.

4.1.3.2 Comparison of the motion resistance distributions across different testing days

Figure 39 compares the distribution of motion resistance values across both days of testing. As mentioned before, there is virtually no difference between the isotropic and anisotropic models. However, this is purely a function of a lack of progressive steering data, and if more data were collected, these distributions may be different. Comparing the motion resistance distributions across both days, we see that the mode values are fairly similar. However, the 17th of October showcases a tighter distribution of values, whereas the 19th of October's results show more spread-out results. The higher variability on the 19th of October may be attributed to the platform traversing between undisturbed areas of the terrain and disturbed areas, significantly impacting the ease with which the platform can move through the soil.

Based on these distributions, we expect the motion resistance parameter to perform well as a parameter for terrain classification. However, this discussion is reserved for the terrain classification section later in this report.

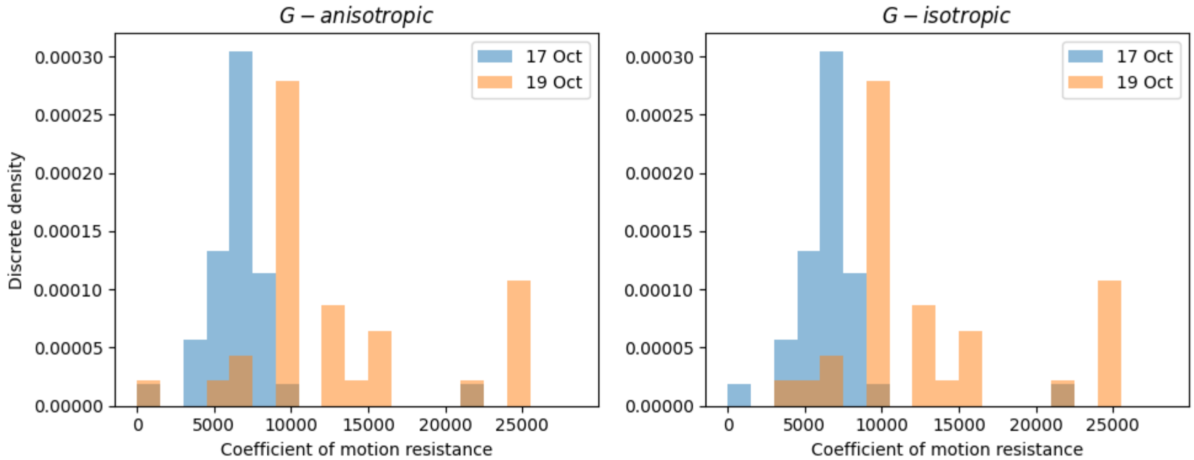


Figure 39 Motion resistance distribution comparisons between the 17th and 19th of October for the isotropic and anisotropic models.

4.1.3.3 Motion resistance value over x-y site map

Figure 40 illustrates the motion resistance values as a function of their location over the test surface. For both days, motion resistance increases as the platform finishes its trip down the ramp. However, once the platform has oriented towards the gate, motion resistance values decrease again. Finally, motion resistance values increase once more when the platform nears the gate (where a slight ramp exists). When comparing the motion resistance trajectories on the same global plot, we find good correspondence between the surface location and the motion resistance. We may, therefore, conclude that the motion resistance coefficient is quite well correlated with terrain conditions and will likely be quite useful as a terrain detection parameter.

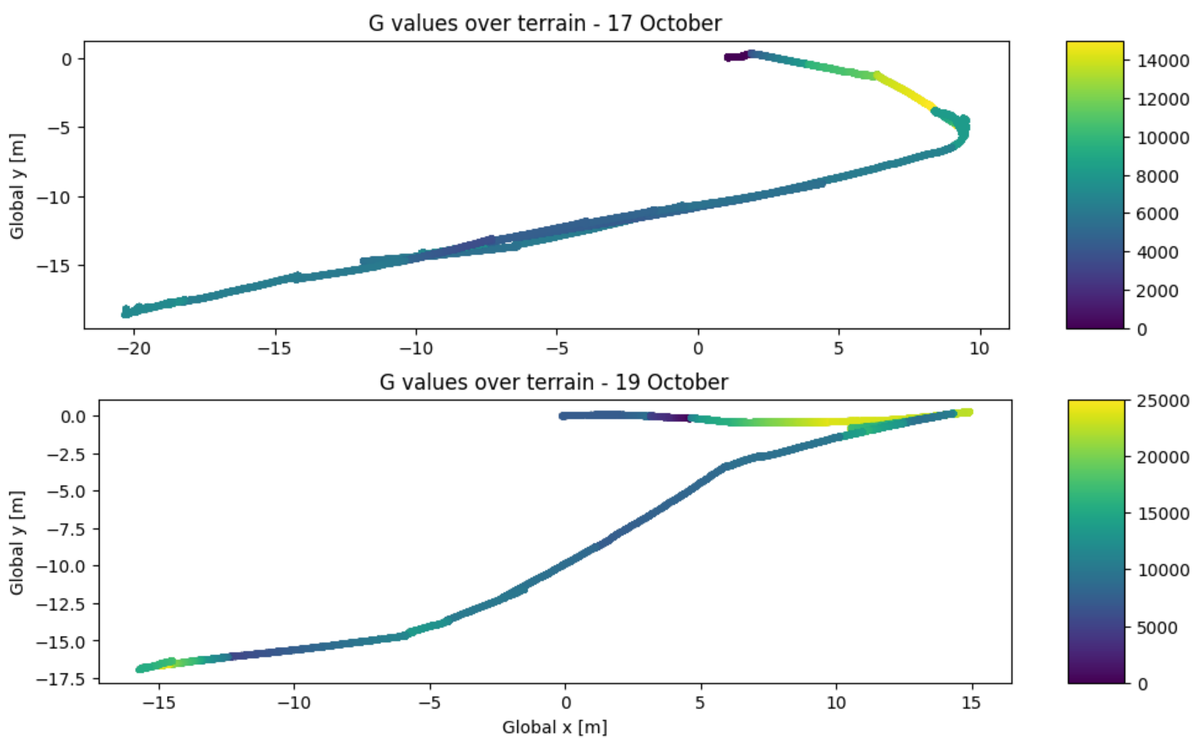


Figure 40 Motion resistance values over the global terrain. The top figure represents the 17th of October in isolation, the middle figure the 19th of October in isolation and the bottom figure both plots superimposed across the same global test site.

4.1.4 POWER MODELLING RESULTS

4.1.4.1 Power modelling accuracy between isotropic and anisotropic models

Figure 41 illustrates the measured and modelled power values for a portion of the run on the 17th of October. We will not report the 19th of October's results because there is nothing to learn from an additional day's comparisons. For this figure, two important results are observed. Firstly, the modelled power often fits the measured data quite well, indicating that the current set of power equations is doing a sufficient job of modelling the measured power. In some areas, the modelled and measured power values are quite different. This occurs when the movement does not meet the algorithm's filtering standards, and therefore, we do not update the parameters, leading to an underestimation (or overestimation) of power. This issue may be solved with more robust filtering rules, however, for the purposes of this project, the current filtering rules worked well for most of the run.

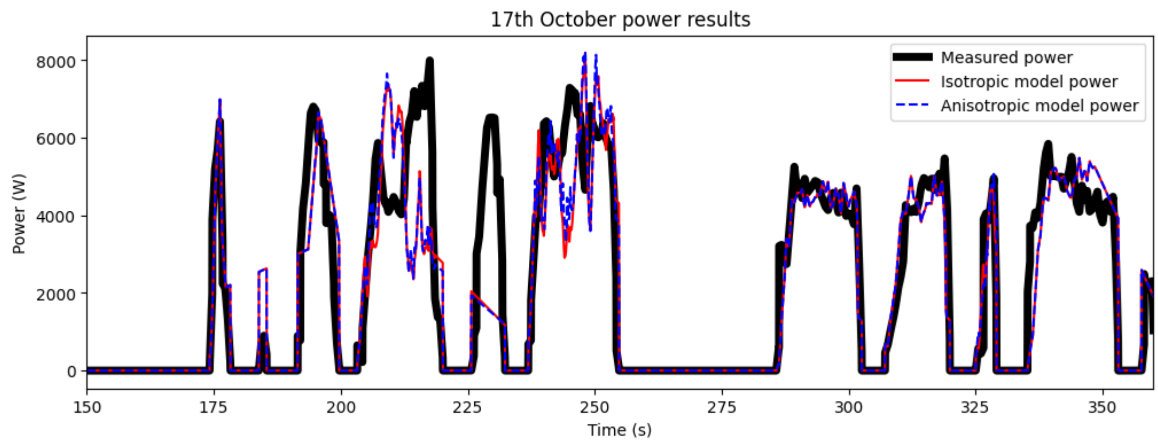


Figure 41 Power modelling results from the isotropic and anisotropic models.

The second finding is that the isotropic and anisotropic models do not have the same modelled power values for certain regions. This is expected because, during skid movements, the anisotropic model has two parameters to aid in the fitting process, whereas the isotropic model only has a single value. It is important to acknowledge that there were limited occasions where the traction values were updated. Therefore, there are large regions of measured power where the traction parameters were not altered. Despite the sparse updates, the modelled power still did a very good job of fitting to the measured data. This indicates that although the traction and motion resistance values are only periodically updated, they result in accurate parameter estimates, which estimate the correct power values in regions without optimisation.

4.1.4.2 Algorithmic loss comparison between isotropic and anisotropic model types

The loss between the measured and modelled power values can be plotted to properly evaluate the difference in modelled power values between the isotropic and anisotropic models. This is illustrated in Figure 42. For both days' testing, the anisotropic model always outperforms or matches the isotropic model. This aligns with expectations, as the anisotropic model has more degrees of freedom to fit the data.

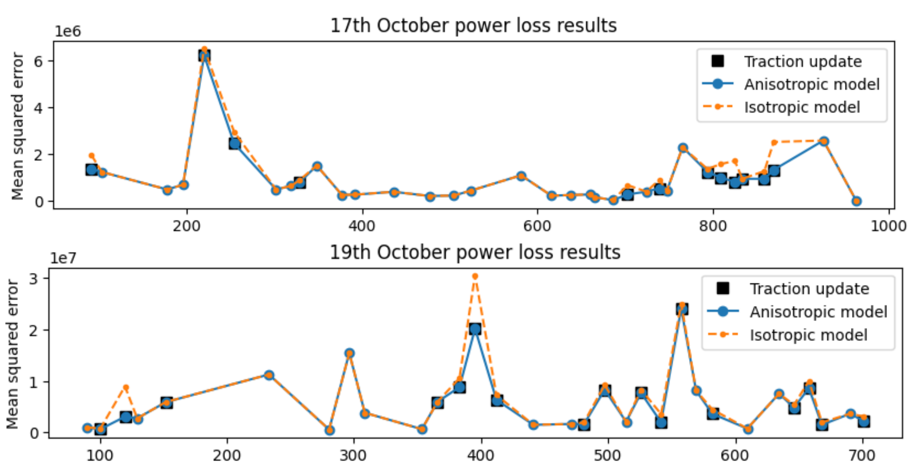


Figure 42 Mean squared error power loss between the isotropic and anisotropic models for both days of testing.

4.2 SLIP PREDICTION ANALYSES

The following sub-section will evaluate the slip prediction model's ability to predict potential slip events correctly. We will illustrate locations where we measured slip (as described in Section 2.4.1) and compare whether the slip prediction algorithm would have predicted these slip events. Slip events will also be evaluated on a terrain map to identify possible insights.

4.2.1 VISUALISATION OF SLIP LOCATIONS ON TIME DATA

Figure 43 showcases the implemented slip prediction algorithm on all data (split across three figures) for the 17th of October. Similarly, Figure 44 illustrates the implemented torque control digital twin on all data (split across three figures) for the 19th of October.

Recall that only straight-driving events are considered for potential slip warning events. To properly evaluate the sensitivity of the proposed torque thresholds outlined in Section 2.4.1 , three warning zones will be defined:

- Warning zone 1 (yellow): 80% of the maximum allowable torque value as per Section 2.4.1. This represents a conservative estimate, which would rather risk early warning, impacting productivity, than run the risk of slipping.
- Warning zone 2 (orange): 100% of the maximum allowable torque value as per Section 2.4.1. This represents the theoretical limit, and acts as a balanced algorithm which warns the user when slip should theoretically occur.
- Warning zone 3 (red): 200% of the maximum allowable torque value as per Section 2.4.1. This represents an algorithm which requires high certainty that slip will occur, avoiding unnecessary warnings, but also potentially missing slip events.

The supplied wheel torque is illustrated as a grey dashed line, with slip events being indicated by green triangles. The true slip events have been highlighted and labelled using red rectangles for easy reference. Before analysing the results, we would like to point out that a few slip events were not highlighted because we deemed them data anomalies. These are slip events where a single slip event was seen for a single wheel. For events where multiple slip events were seen, or slip events were seen across multiple wheels, we clearly highlighted them using the red rectangles.

When analysing the results, we are looking for slip events that occur inside or above the three warning zones, as this means the proposed algorithm (at various sensitivities) is quite accurate at predicting when slip will occur. We fail to predict a dangerous event if a slip occurs beneath these warning zones. There are regions where the measured torque exceeds the least conservative warning zone (zone 3), without slipping. In these instances, the algorithm was likely still too conservative. However, this is not a bad property, as we would rather incorrectly predict a slip event than miss a slip event. Recall that currently, CMTI has no slip prediction algorithm in place, and therefore, even a single correctly predicted slip event would mean the digital twin is adding value.

With these insights, the data can be analysed. Starting with the 17th of October's run, the first slip events occur at labels 1 and 2. Although the supplied torque is in or near the warning zones, the slip events are likely not a function of the applied torque, but rather due to the movement thresholding process when splitting linear movements from progressive movements. The slip events are at the cusp of the transition, meaning the slippage is likely due to a progressive steer movement rather than straight-line wheel slipping. We ignore these slip events, as the operator likely planned them.

At label 3, two reverse movements resulted in slipping. When reviewing the video footage, these movements occurred during a phase where the boom scraped on the ground, causing the rear wheels to lose traction and slip. The slip events in this region are not attributed to terrain properties, but to an external, unpredictable event. We may, therefore, ignore these results because unrelated factors caused them.

Label 4 shows multiple slip events for both rear wheels, indicating continued slipping. The algorithm correctly predicted these events, with the measured torque in the second and third warning zones. At label 5, we see a singular slip event, which may likely be ignored, but we highlight that it was measured in the second warning zone. Therefore, we would have successfully predicted it if it were a short slippage event. At label 6, continued slipping occurs for a long period for both rear wheels and, for certain periods, the front wheels. Although the front wheel torque values are well within their envelope and weren't predicted to slip, we believe the loss of traction on the rear wheels caused the front wheels to slip.

When looking at the rear wheel data, all slippage events occurred in the second warning zone, confirming that the digital twin would have prevented these events.

Labels 7, 8, 9, 10, 11 and 12 present point slip events, sometimes near the cusp of movement type transition and may likely be ignored. However, it is highlighted that the slip events are all found within warning zone 2; therefore, if any of these were not data anomalies, they would have been prevented successfully.

Label 13 shows a set of slip events for the rear right wheel. In this scenario, the slip occurred in warning zone 3, implying they would have been successfully predicted ahead of time for both warning zones 2 and 3.

In summary, all the serious slip events (non-point events) that occurred, were always found above warning zone 2, meaning the user-defined safety zone was not necessary, and all serious slip events could be predicted using the model parameters alone. This indicates that the conversion from traction estimate to torque limit requires no additional calibration.

Another key conclusion from these results is that the front wheels never slipped on their own, serving as an additional validation that the CMTI platform is quite front-heavy, and the rear wheels are expected to slip long before the front wheels. Looking at the results, the front wheels often have a large envelope of torque available to them, where the rear wheels are usually at or near their limit. We, therefore, recognise that based on the accuracy of the proposed digital twin's torque envelope estimations, we can likely extend the current work towards a torque balancing algorithm, where the torque at each wheel is varied to minimise the risk of slipping at any given wheel. However, such an algorithm would require CMTI to incorporate the proposed digital twin into their control software, before complete validation of the torque control algorithm is possible.

The focus is now shifted towards the analysis of 19 October, where both the surface conditions and machine were measurably different. If the proposed algorithm succeeds at the slip prediction task under this scenario, there is initial evidence that the proposed slip prediction algorithm can generalise to different assets and terrains.

Once again, point slip events are ignored, and only the relevant slip events are highlighted. Starting with label 1, all four wheels slip. However, all four slip events are well-predicted by the algorithm for warning zone 2. Label 2 and 3 likely shows slip events due to progressive steering. However, these slip events are once again detected in the second warning zone, indicating a well-calibrated algorithm. Label 4 represents an event where the rear right wheel experienced significant slip. The proposed digital twin successfully predicted this event above warning zone 3. This implies warning zone 2 would have been too conservative, but still correctly predicted the slip event.

Labels 5, 6 and 7 again likely represent slip events due to progressive steering; however, if they were true straight-line slip events, they were successfully identified within the second and third warning zones. Finally, label 8 represents another event where the rear right wheel experienced significant slip. The supplied torque was found within the second warning zone, showcasing the algorithm's ability to predict slip events ahead of time.

In conclusion, the proposed slip prediction algorithm is fairly accurate at predicting slip events, and should perform well at preventing these events. The results showed that most slip events occurred during or above the second warning zone. Therefore, when the digital twin is implemented in production, the maximum allowable torque should be calculated based on the method outlined in Section 2.4.1, to ensure the majority, if not all of the unplanned slip events are avoided.

Such an implementation may come at the cost of slightly lower wheel torque values and should be fine-tuned to each application's needs. If productivity is a high priority, with minimal safety concerns, a high torque threshold can be chosen (like warning zone 3). However, a conservative threshold like warning zone 1 can be selected if safety is an immediate concern. The default warning zone 2 should suffice for a strategy between the two extremes.

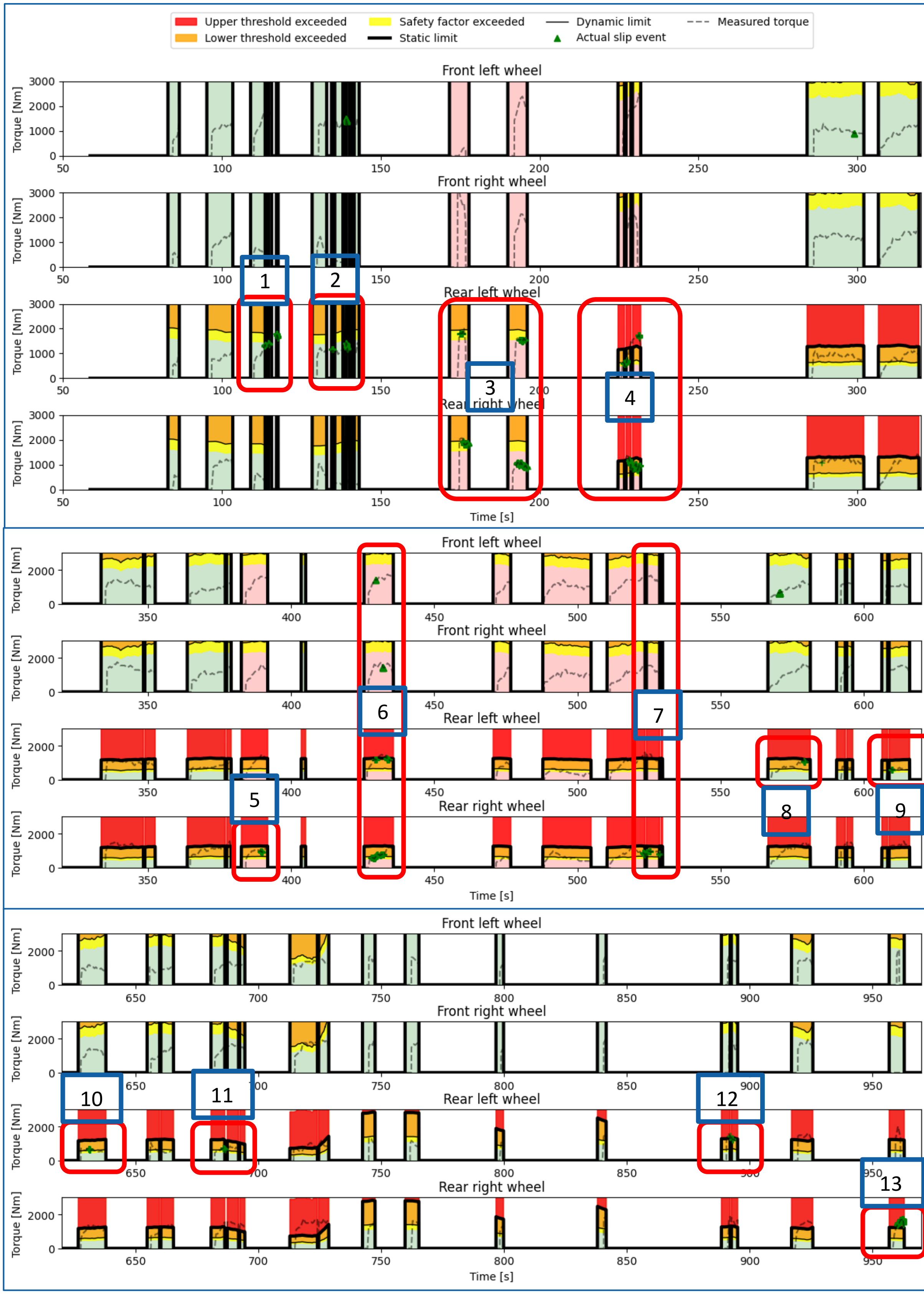


Figure 43 Measured torque, predicted torque envelope and slip events for each wheel during 17 October run.

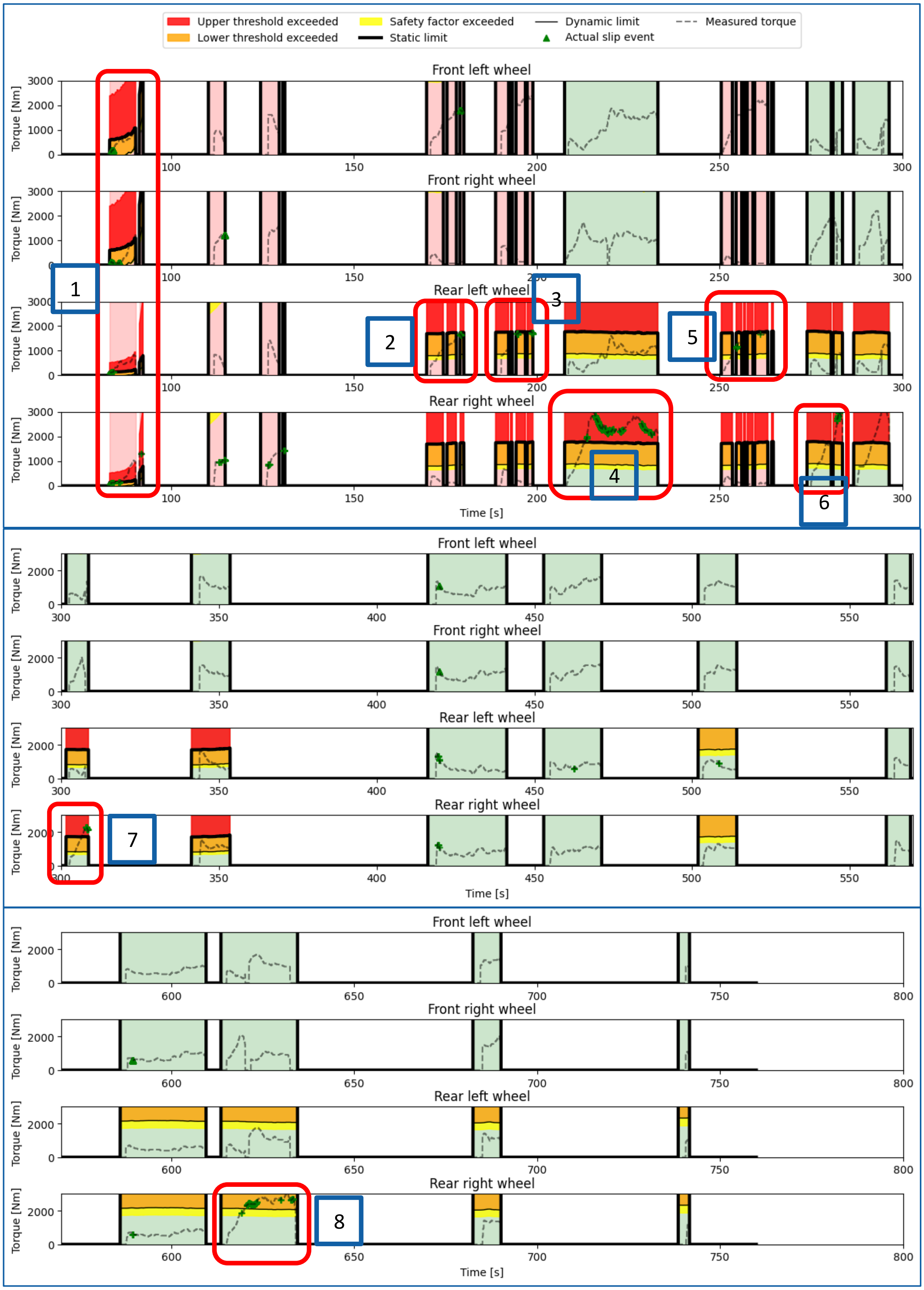


Figure 44 Measured torque, predicted torque envelope and slip events for each wheel during 19 October run.

4.2.2 VISUALISATION OF SLIP LOCATIONS ON X-Y TERRAIN PLOT

To better understand the traction of the platform's wheels, all slip events have been plotted for all four wheels on the 17th and 19th of October in Figure 45 and Figure 46, respectively. For the 17th of October, we see a few isolated slip events for the front wheels, which can be ignored. The rear right wheel was more prone to slipping, which could indicate a slight imbalance in the lateral COM location, with a bias towards the left of the platform. The rear left wheel seemed more likely to slip under reverse movements, whereas the right wheel showed no specific bias. For the 19th of October, we also see a few isolated slip events for the front wheels, which may be ignored. Once again, the rear right wheel was more inclined to slip, which points towards a left-biased COM. Interestingly, the rear left wheel was more prone to slippage during reverse movements, with the rear right wheel slipping in the same region during forward movements.

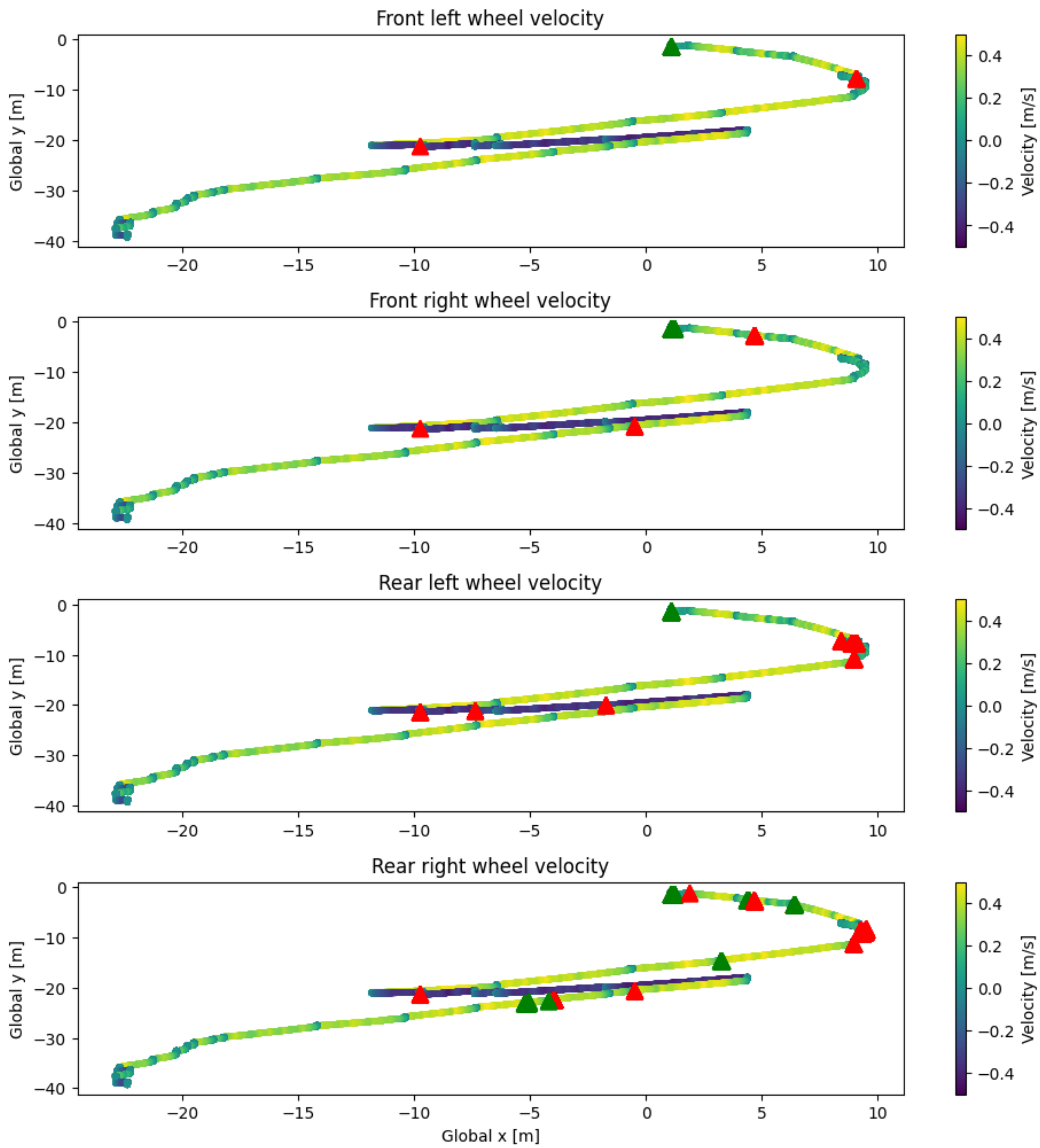


Figure 45 Slip events for all four wheels during forward (green) and reverse (red) movements on the 17th of October.

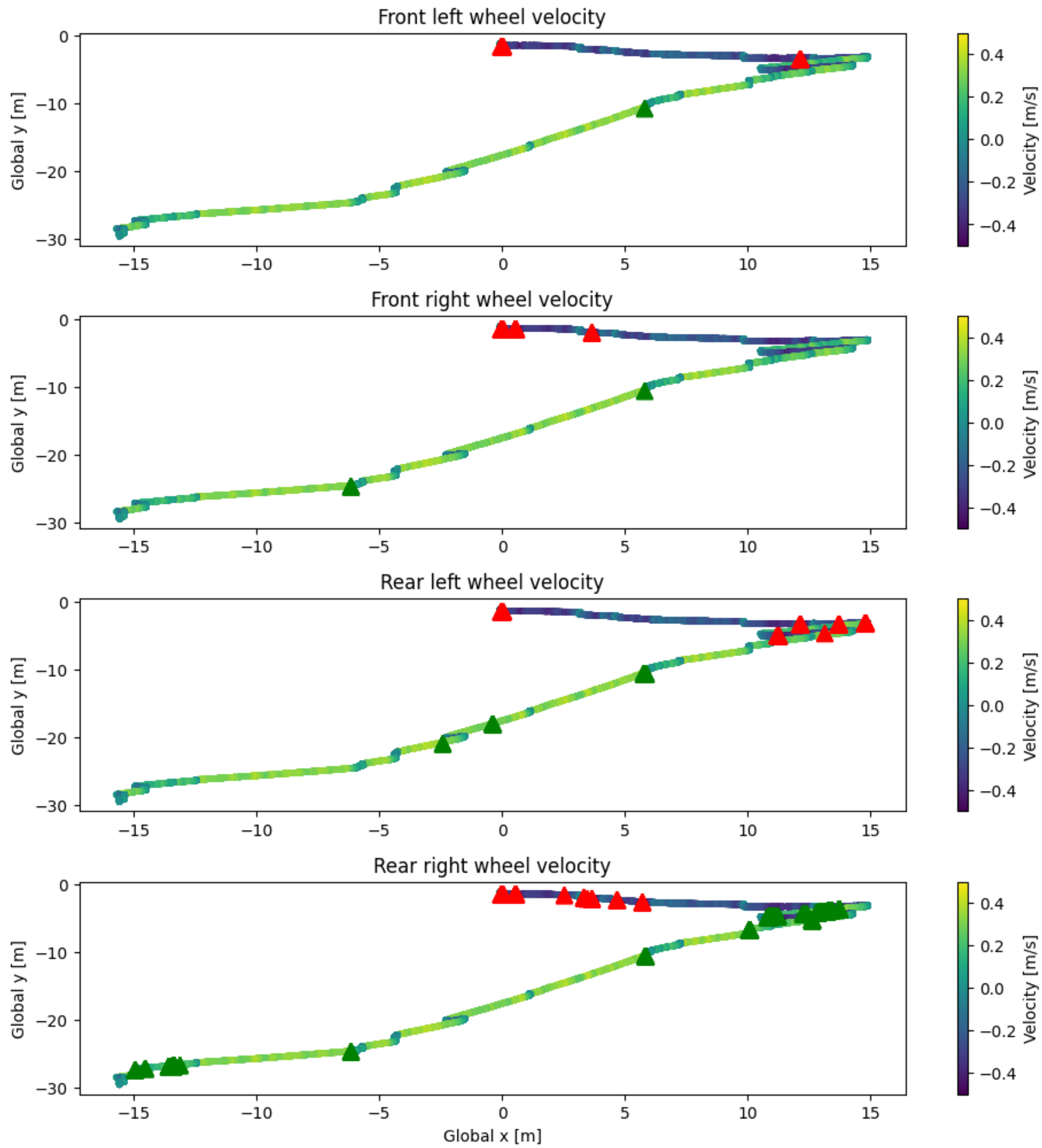


Figure 46 Slip events for all four wheels during forward (green) and reverse (red) movements on the 19th of October.

In conclusion, we have further evidence that the platform has a COM bias towards the front, but also likely slightly to the left, meaning the rear left wheel is the most likely to slip. Based on this finding, we recommend the CMTI platform rather skid/progressive steers towards the right, as the left wheels set likely has slightly less weight to counteract rotational motion. Right rotations should, therefore, use up less power.

4.2.3 COMPARISON OF SLIP LOCATIONS ACROSS SITES

As a final measure of terrain conditions, we may plot the slip locations for both days' testing for all four wheels (Figure 47). This should reveal any areas of the terrain that were especially prone to wheel slip. Looking at the rear wheels, we see that near locations with sharp turns, wheel slip is soon to follow. This makes sense, as the turning movement disturbs the soil, causing terrain conditions with less traction in the longitudinal direction, leading to slip events.

Areas without turning-induced slip are seen near the end of the run (-10; -15). Here, we had an area that was much muddier than the rest of the terrain, and therefore, the wheel slip across both days is sensible. Finally, we see some slip events near the gate, which were due to small movements that caused the platform to sink into the soil, leading to a lower traction scenario.

In summary, the platform was seen to be more likely to slip during straight-line movements if a skid/progressive steer movement preceded these movements. We also see an overlap of low traction near the end of the run due to muddy conditions across both days.

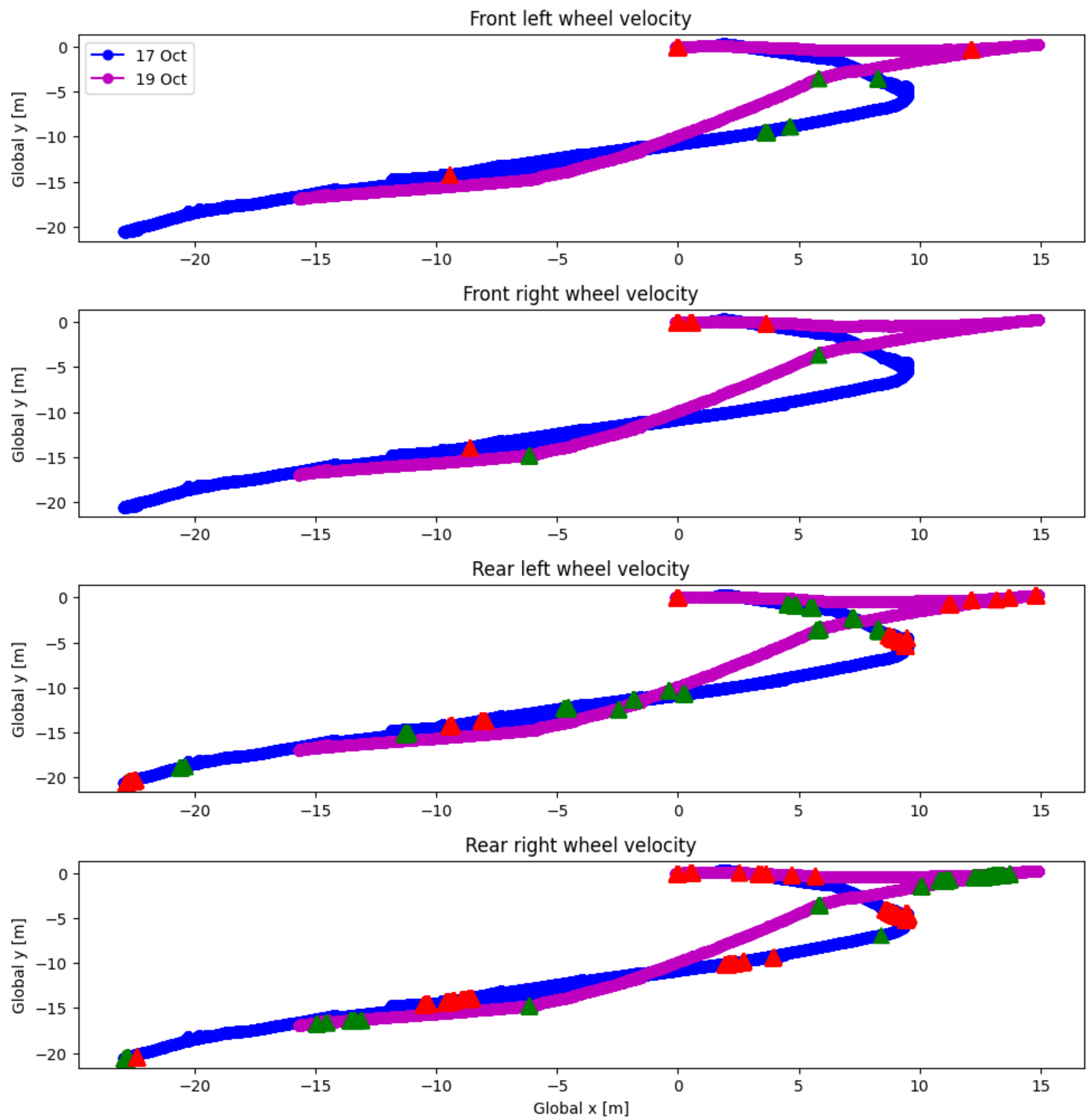


Figure 47 Comparison of wheel slip locations for both days' testing.

4.3 TERRAIN-CHANGE DETECTION ANALYSES

The terrain change detection algorithm represents the second part of the digital twin that creates actionable insights. The following sections will analyse the terrain parameters, their relationship and ultimate their ability to be used as inputs for terrain classification. Recall from Section 4.1.4.2 that the anisotropic model was shown to outperform the isotropic model (at least for parameter calibration) and therefore we will no longer refer to it during the following analyses.

4.3.1 ANALYSIS OF THE RELATIONSHIP BETWEEN TERRAIN PARAMETERS

Figure 48 and Figure 49 illustrate the statistical relationship between traction model parameters across both days of testing.

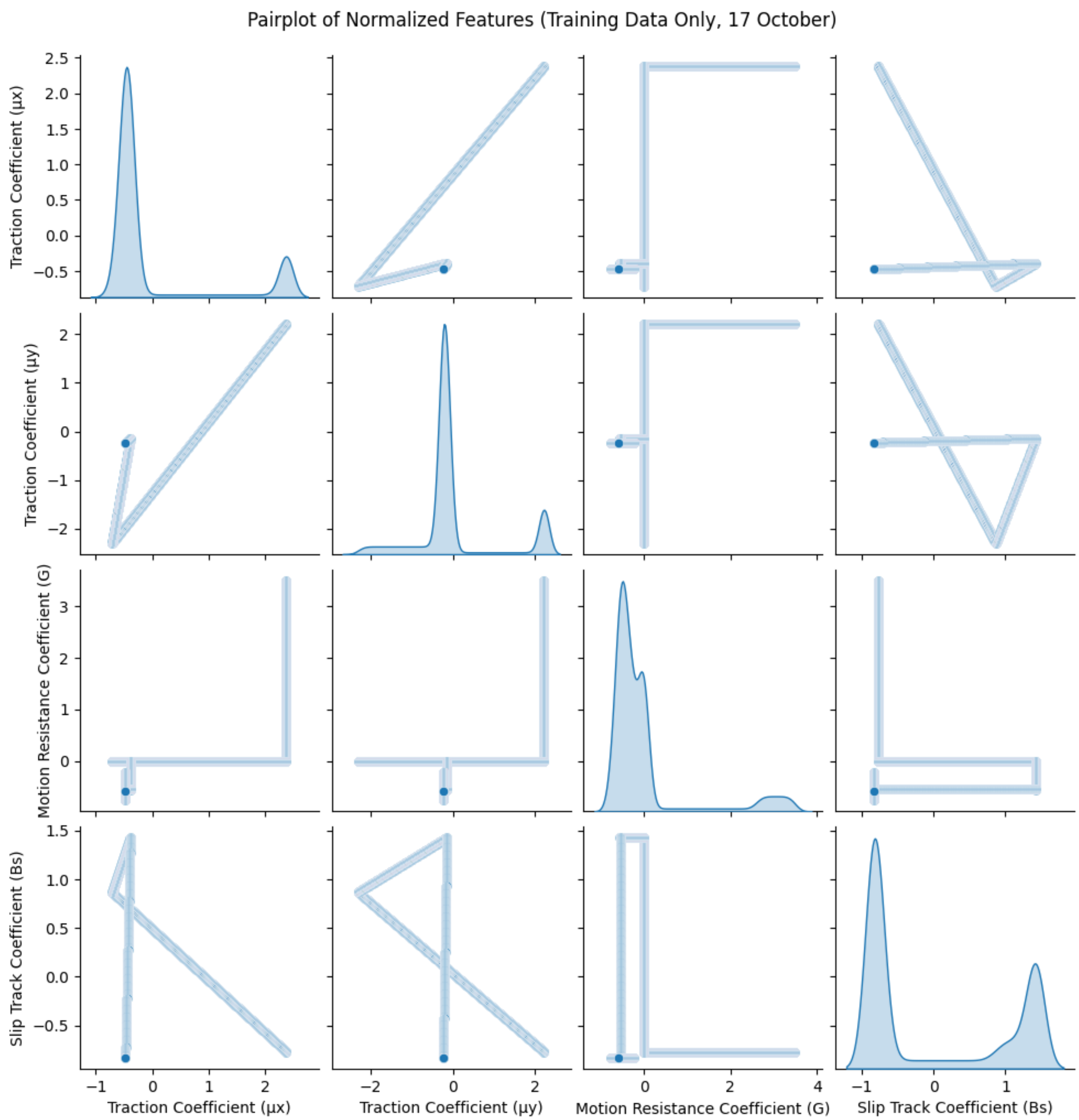


Figure 48 Relationship between traction model parameters for the 17th of October.

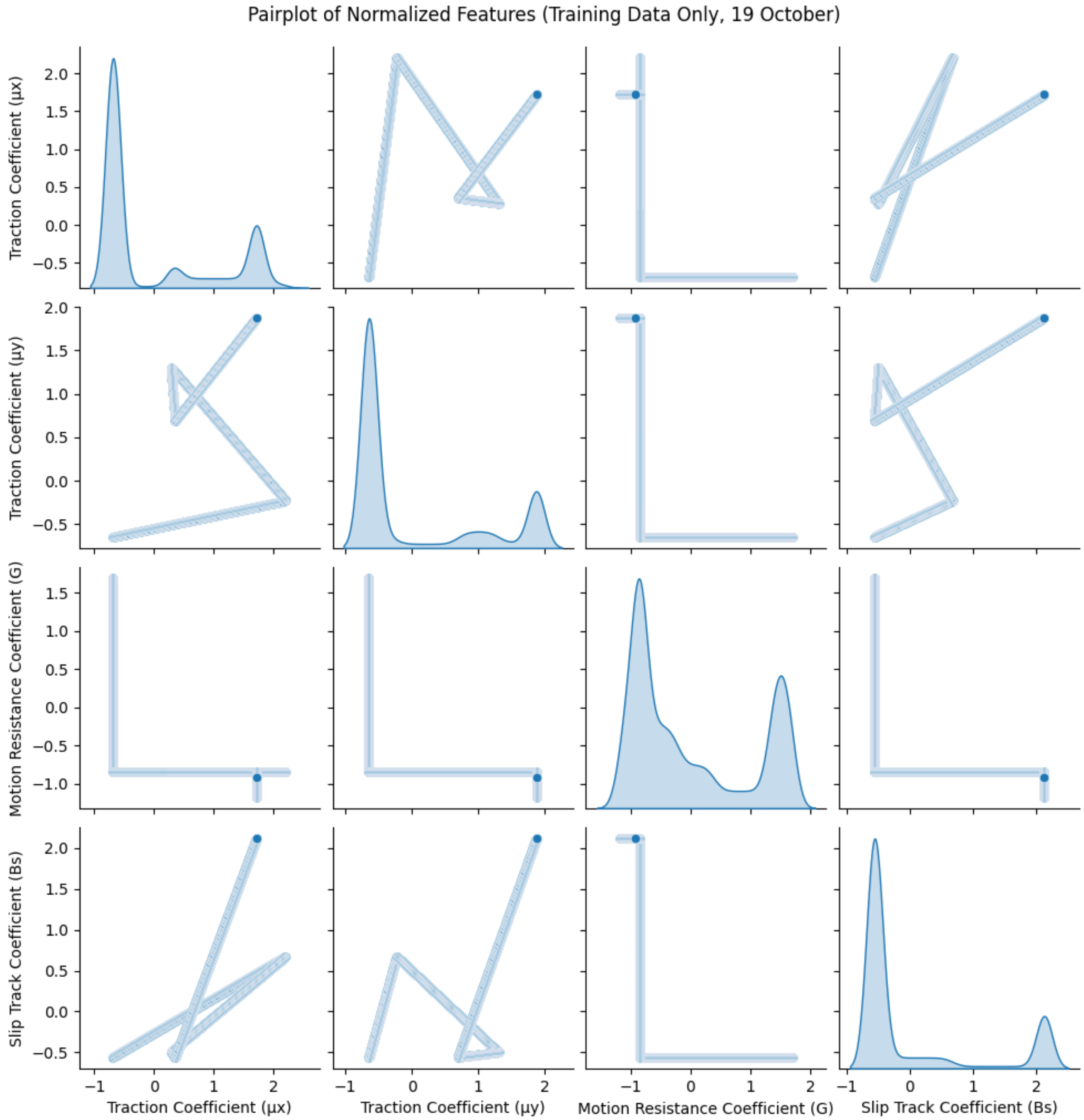


Figure 49 Relationship between traction model parameters for the 19th of October.

For both the 17th and 19th of October, parameters tend to be bi-modal, with the left mode more dominant than the right mode. This would indicate that constant/near-constant values are estimated for most of the run, with a few outliers creating the second mode.

When the relationship between the traction coefficients is compared, a somewhat correlated positive linear relationship results, although it is not always the case. When comparing both days, the slip track does not seem well correlated with the traction coefficients. On the 17th of October, a negative correlation is found, whereas a positive correlation is found on the 19th of October.

The motion resistance coefficient has almost no correlation with the other parameters, mainly because it is usually updated when the other parameters remain fixed (straight-line driving) and vice versa.

The results are quite sensible, but it is difficult to estimate whether the parameters can successfully classify terrain conditions. The following section explores this question using the one-class SVM classifier.

4.3.2 TERRAIN CLASSIFICATION PARAMETER SENSITIVITY

Determining which terrain parameters are most important when attempting terrain classification is useful. Therefore, we have tested various parameter combinations for the anisotropic model and observed the resulting model outlier detection performance. The result of this test is illustrated in Figure 50.

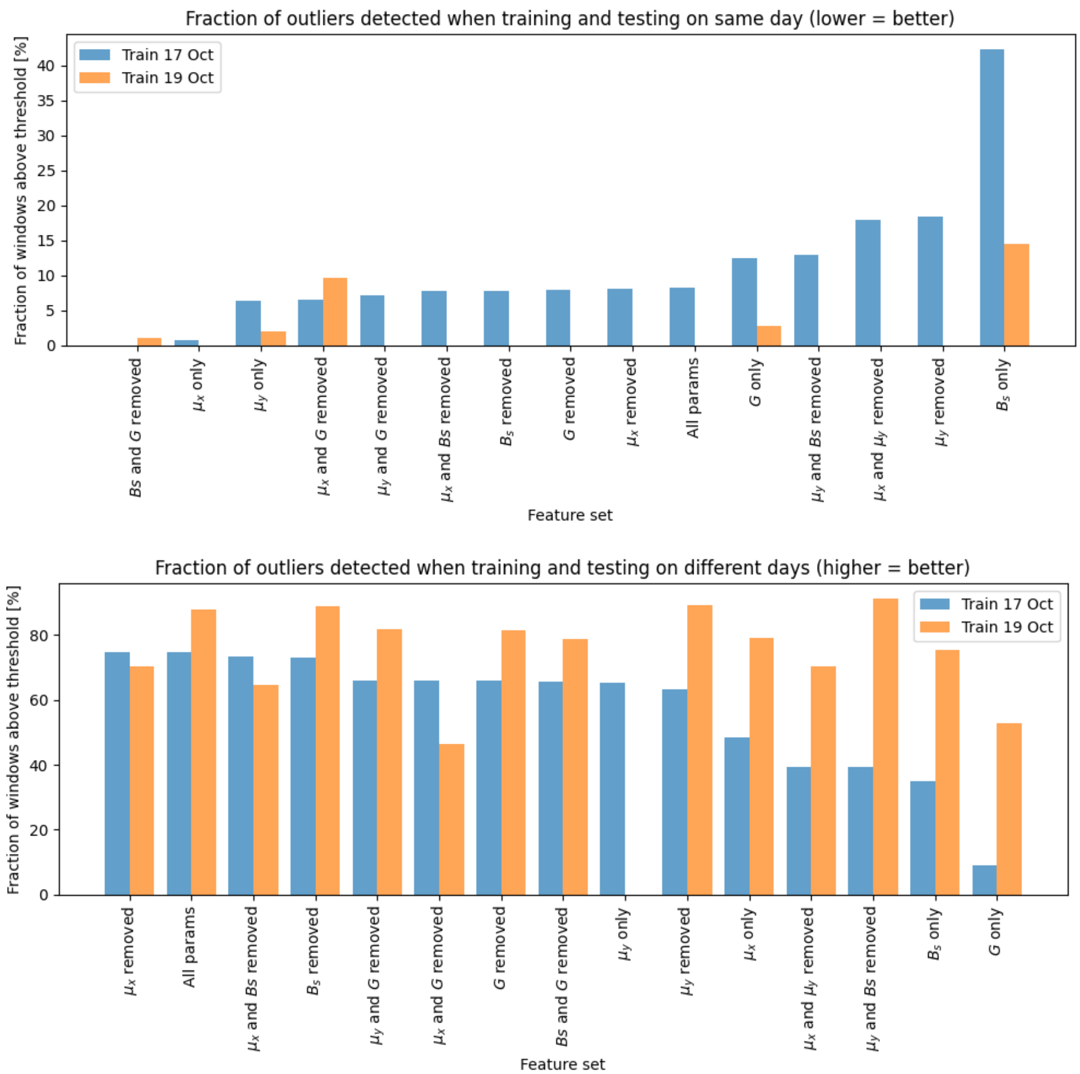


Figure 50 Terrain classification performance of SVM model using various combinations of model parameters as training data.

For the top figure, we plot the % of data points that were identified as outliers when testing on the same day's terrain as the testing set. Here, we expect lower scores to be better. For the bottom figure, we plot the % of data points that were identified as outliers when testing on the opposite day's terrain as the testing set. Here, we expect higher outlier scores to be better because the terrain between the 17th and 19th of October is known to be slightly different due to rain.

For the top plot, we find numerous parameter combinations that outperform the "all parameters" equivalent. However, in the bottom plot, we see that those scenarios' accuracy at identifying outlier terrain is less than the "all parameters" equivalent. Therefore, the models that outperform the "all parameters" model at same-day outlier detection have overfitted to this scenario and are not useful at detecting different-day outliers. The "all parameters" model is the only model that performs consistently well across both tasks and is therefore seen as a balanced model.

Analysing the top figure, we can evaluate two extremes: Using a single parameter and removing a single parameter. We see high same-terrain misclassification rates using only the slip track or motion resistance coefficients. However, μ_x and μ_y are quite descriptive of the terrain, having few misclassification errors.

If we look at the removal of a single parameter from the parameter set, we see that Bs, G and μ_x have little effect on the accuracy, however, μ_y has a large impact, making the final model inaccurate. Therefore, we may conclude that μ_y plays a significant role in identifying whether the current terrain has changed.

We may ask the same questions in the scenario where we test on different terrain to the training data. When using Bs or G in isolation, we see poor identification of terrain change. In isolation, μ_x performs slightly better. However, μ_y seems to completely fail on the 19 October training scenario, meaning it is not a reliable parameter to use in isolation for terrain change identification. When investigating the impact of parameter removal, we see that the removal of μ_y has little effect on the accuracy scores, supporting the above statement that μ_y is not a reliable parameter for terrain change identification across different scenarios. The removal of G leads to slightly lower accuracies, indicating that G likely has some meaningful effect on estimating terrain change over different days. The removal of μ_x and Bs have little impact on the overall top accuracy.

Looking at all these interpretations, the following conclusions can be drawn:

- The slip track seems to have little impact on same-terrain and different-terrain outlier detection. It can likely be removed from the terrain classification task with little impact on model accuracy.
- The motion resistance coefficient seems to have a larger impact when describing a difference in terrain compared to the same terrain. Therefore, it is an important parameter and recommended to be included in the terrain classification task, as it helps the model differentiate a change in terrain when there is a real change.
- The longitudinal traction coefficient significantly impacts the accuracy of terrain classification for same-terrain and different-terrain scenarios. From the results, this was the single parameter with the largest impact on model performance.
- The lateral traction coefficient significantly impacts the model accuracy for same-terrain identification. However, it has little impact when looking at the results for different terrain classifications.

Therefore, each of the parameters, besides perhaps the slip track, increases the final model's accuracy. Some reduced models outperform the all-parameter model in one task but perform poorly in the opposite task. Therefore, each parameter plays a role in creating a final model that performs well on both tasks.

From the aforementioned conclusions, we have also decided to remove Bs from the model, and evaluate the improvement, which is illustrated in Figure 51. We see that the model without the slip track is more stable and is, therefore, the final refined model we recommend. The following section showcases some final results using this model.

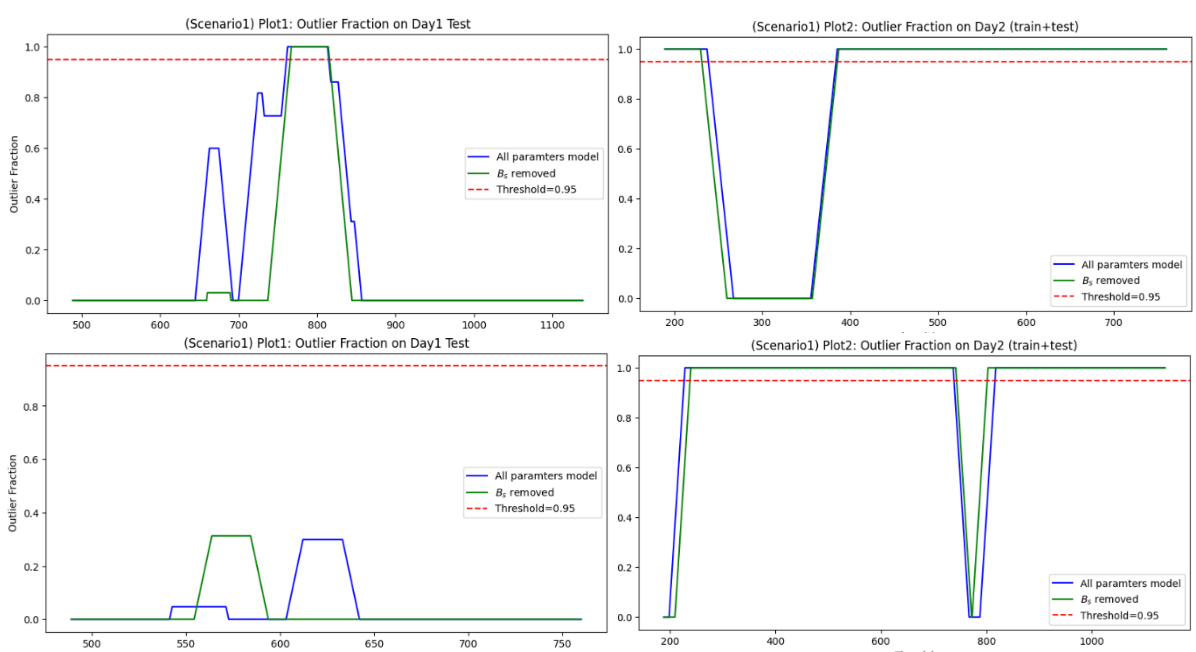


Figure 51 Once class SVM performance when being trained and tested on various days' data using only μ_x, μ_y and G as training data.

4.3.3 TERRAIN CHANGE IDENTIFICATION SCORE ACROSS TERRAIN

The results of the final one class SVM classifier is presented in Figure 52.

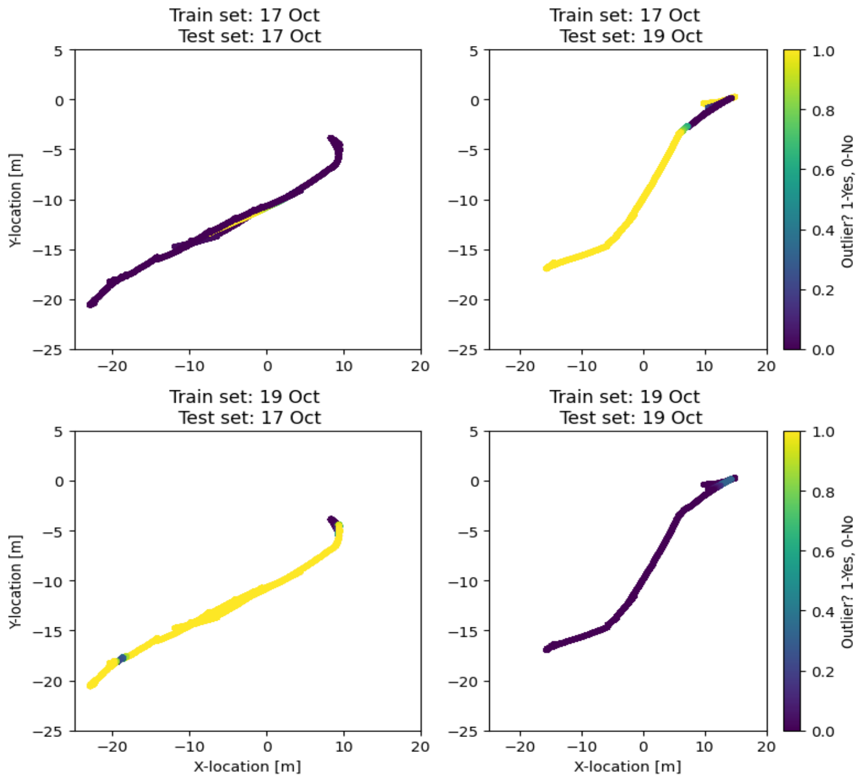


Figure 52 Terrain classification results visualised over the x-y global coordinate system for both days of testing.

The top left figure represents the scenario where we train on the first portion of 17 October data (using only μ_x, μ_y and G), and test the model on the remaining data. We see that for most of the run, except for a small portion in the middle, we successfully identify the data as not being an outlier. In other words, the algorithm correctly identifies that the terrain has not changed significantly from the training set. The top right figure represents the same case but tested on the 19th of October's data. In this scenario, we detect high outlier scores for most of the run, indicating that the model identifies the different day's test data as different from the 17th of October. This implies that the model is successfully detecting the changes in terrain conditions between the 17th and 19th of October. Considering the terrain was muddier on the 17th of October, these results make sense and showcase a terrain detection model that is functioning well.

Analysing the opposite case, namely training on a portion of 19 October's data, reveals similar results. The fact that we can train on any of the two days shows that the algorithm is not overfitting to a specific day's data, but is learning some underlying distribution representing terrain properties.

In conclusion, the proposed one-class SVM model can detect terrain changes and be deployed alongside that slip prediction algorithm as an added insight at no additional technological cost. The data input to the SVM originates from the traction estimation process. The fact that the one-class SVM model could correctly identify changes in terrain also implies that the traction model parameters are quite meaningful and useful in describing the terrain properties.

5 CONCLUSIONS

This report summarises the development and validation of a traction estimation digital twin tailored to skid-steered vehicles, particularly aimed at predicting when wheel slip may occur and detecting when the wheel-terrain relationship has changed.

Section 2 described the construction and foundational development of the traction estimation digital twin, focusing specifically on the physical entity, the processes for physical-to-digital updating, the digital models, the digital-to-physical updating algorithms, and the continual optimisation of these twinning dimensions.

The physical entity dimension outlined the CMTI roof bolter platform as the relevant entity. It was determined that platform orientation, rotation rate, position, and velocity could not be obtained directly from the onboard systems, necessitating additional sensor integration. Two sensor suites were tested with a single LIDAR sensor and a tilt sensor, demonstrating comparably accurate results to those from a suite comprising two tilt sensors, a gyro, a ground speed sensor and a GPS. Due to the slightly lower accuracy of the latter option, the LIDAR-tilt suite was selected for the remainder of the project. The physical-to-digital updating dimension outlined a Kalman filter to filter all sensor data, generating filtered estimates for the platform pose, velocity and turning speed. The digital model dimension specified two models, namely a wheel force estimation model and a traction estimation model. The wheel force model was developed in Simscape and was used to build a force lookup table that takes the filtered platform pose and convert's it to the corresponding wheel forces. The traction estimation model used a model of platform power to estimate the underlying terrain properties. Previous works used an isotropic traction assumption for the power model. The current work extended existing works by developing a novel anisotropic traction model which can distinguish between longitudinal and lateral traction forces. The digital-to-physical updating dimension converted the traction estimates to actionable insights using slip prediction and terrain change detection algorithms, completing the digital twinning loop.

Section 3 detailed the calibration process of the sensors, models and algorithms used in the digital twin. The calibration process showed that platform pose, turning rate, position and velocity could be accurately measured by Kalman filtering the measurements from the LIDAR and tilt sensors. Lifting tests were used to calibrate the mass and centre of mass values of the platform, which were sanity-checked and representative of observed platform behaviour. The tests revealed a front-heavy platform, meaning the rear wheels always skid long before the front wheels. This behaviour was seen from the data and video footage, confirming that the centre of mass calibration process was correct. The mass values were then decomposed into their fundamental parts to build a wheel force estimation model in Simscape. The model was validated against analytical formulas, instilling confidence in the force estimates. Finally, a pull test on the surface revealed that the estimated longitudinal traction values aligned with measured values, instilling confidence in the anisotropic traction estimation algorithm.

The slip prediction algorithm required threshold calibration to estimate when real slip events occurred. A reliable threshold was determined using video footage and differences in wheel and platform speeds. A parameter search study calibrated the terrain change detection algorithm, revealing optimal parameters that performed well.

Section 4 implemented and evaluated the digital twin through comprehensive surface tests. These tests investigated the three main digital twin algorithms: traction estimation, slip prediction, and terrain-change detection.

The anisotropic traction estimation algorithm yielded sensible results. The isotropic equivalent traction parameter was bounded below and above by the longitudinal and lateral traction coefficients from the anisotropic model, respectively. This showcased that the novel anisotropic formulation was correct and that the lateral traction was higher than the longitudinal traction. This finding indicates that the lateral resistance plays a much larger role during skid or progressive steer movements compared to the longitudinal resistance.

The ratio between the lateral and longitudinal traction coefficients, the traction ratio, was shown to be a useful parameter for the quantification of turning resistance. The higher the ratio, the more power will be expended to complete a turn. This ratio has previously been neglected in literature as the isotropic assumption, per definition, assumes this ratio is unity.

The slip estimation algorithm performed well at predicting when a wheel would slip. This prediction depends on the estimated terrain traction, wheel normal force and platform pose. The fact that the slip prediction performed well served to validate the models, instilling confidence in the proposed digital twin.

The anisotropic traction algorithm parameters were additionally used to detect changes in terrain conditions. Two days of testing were conducted with different terrain conditions, and therefore, sufficient data was available to evaluate the terrain classification algorithm. The algorithm showed promising results, successfully detecting when a change in terrain was present.

The proposed digital twin is a promising first step towards a real-time approach for traction estimation, slip prediction and terrain classification for skid-steered vehicles.

6 RECOMMENDATIONS

Based on the findings of this project, we propose several recommendations to develop and refine the current research. These recommendations span multiple areas, including testing methodologies, algorithm improvements, digital twin development, and practical implementation strategies.

6.1 IMPROVED TESTING STRATEGIES

Future testing should prioritize a scale model or a pre-production prototype rather than a production vehicle. Coordinating tests on a vehicle being prepared for underground operations proved to be logistically challenging and time-restrictive. Future experiments should also include pull tests in both longitudinal and lateral directions to experimentally validate traction coefficients across different movement axes. Finally, all future experiments should incorporate video recording, which has proven invaluable for sensor synchronization and data validation.

With knowledge of which sensors work well for the various algorithms, an additional set of experiments would be helpful to determine:

- The minimum necessary set of sensors for reliable traction estimation.
- The most cost-effective sensor combination.

We further recommend that research be conducted into whether a more cost-effective LIDAR sensor could achieve similar traction estimation results, particularly at low speeds, to determine if additional sensors are necessary.

6.2 ENHANCEMENTS TO DIGITAL TWIN MODEL ELEMENTS

Future work should explore whether the computational overhead of Simscape can be reduced, allowing for real-time simulation of dynamic roll and pitch effects without relying on precomputed lookup tables. This would allow dynamic forces to be incorporated instead of quasi-static forces.

The traction estimation algorithm should transition from an optimization-based approach to a Kalman filter-based approach. While Kalman filtering is computationally complex, it could significantly improve accuracy, as the optimisation-based approach is sometimes quite stochastic. Additionally, the traction estimation model should incorporate a power term to account for acceleration effects, which are significant in large mining vehicles. We presently ignore these acceleration transients, but it would be much better to model for these cases. Finally, traction should be estimated for each wheel independently. This would require improving the ICR estimation method, which currently assumes a tracked vehicle.

We recommend extending the slip prediction algorithm to a torque control algorithm, which must be tested in a controlled environment before deployment in production vehicles. This phase should validate its functionality and effectiveness before considering implementation in a mining environment.

The terrain classification model should be enhanced to improve interpretability. Currently, it can only detect that a terrain change has occurred. Future work should develop a supervised machine learning approach to determine whether the terrain change indicates an improvement or deterioration in conditions. Alternatively, a fleet-based approach should be developed with insights from multiple machines to better interpret outlier values.

6.3 INDUSTRY APPLICATIONS

While many recommendations highlight CMTI, they broadly apply to any mining company interested in implementing and refining the proposed digital twin model. The models and algorithms developed in this project can be replicated and calibrated for different vehicles, offering broader industry adoption potential. Since CMTI was a key research partner, the current models are tailored to their platform, but similar calibration procedures could be applied to other mining vehicles.

6.4 FINAL COMMENTS

We recommend further refinement and expansion of the digital twin, focusing on cost-effective sensor configurations, improved real-time simulation, and more advanced traction and terrain estimation algorithms. Initial testing should be conducted on a smaller, more manageable vehicle to facilitate rapid iteration before scaling to full mining platforms. Additionally, future research should explore alternative sensor solutions, improved machine learning approaches, and better integration of force estimation models. These recommendations will maximize the impact of the digital twin, ensuring that it becomes a practical and scalable solution for improving traction estimation and vehicle control in mining operations.

7 WAY FORWARD

Building on the outcomes of this project, the most effective path forward requires taking a step back to refine and expand simulation capabilities. While developing the force estimation model in Simscape, we successfully created a well-calibrated digital replica of the CMTI platform. This digital model extends beyond its original purpose of wheel force estimation—it can serve as a powerful simulation tool for broader research and development applications.

Given this background, the next logical step is to further the development of the digital CMTI platform within Simscape. A fully calibrated simulation model would enable extensive testing across various operational conditions, removing the need for direct real-world testing on the production platform. "What-if" scenarios can be explored in a controlled virtual environment at a fraction of the cost, offering valuable insights without disrupting mining operations. We, therefore, propose the development of a simulation digital twin.

This digital twin would provide a controlled testing environment, allowing systematic alteration and analysis of various wheel configurations, geometries, materials, motor specifications, gearboxes, dip angles, terrain properties, and more. The resulting behavioural data can then inform vehicle design and mine layout optimization, enabling simulations of mine traffic patterns, terrain interactions, and operational constraints before real-world implementation.

The foundation established in this project, particularly in measuring and modelling terrain properties, will be critical in ensuring physically accurate simulations. A simulation-driven approach will also enhance the ability to conduct algorithm development similar to what was undertaken in this project. By defining terrain conditions within the simulation, we gain access to ground-truth data against which algorithms can be validated. This enables precise performance evaluation of:

- The traction estimation algorithm, by comparing estimated traction values to known terrain properties.

- Various torque control algorithms without the risk and cost of live vehicle testing.
- The terrain change detection algorithm, by testing its ability to identify known environmental changes in a controlled setting.

By integrating these elements, the simulation model will exponentially increase the utility of the current research, extending its impact beyond traction estimation to broader vehicle and mine design applications.

To ensure efficient and effective development of the simulation digital twin, we propose starting with a smaller, simpler physical test vehicle—a remote-controlled skid-steer robot—before scaling up to the full CMTI platform. This approach offers several advantages:

- Faster, cheaper, and more controlled experimentation – A small in-house vehicle allows for rapid calibration and iteration without the logistical and financial constraints of full-scale testing.
- Minimized production disruptions – Unlike the CMTI platform, which is built for active mining operations, a dedicated experimental vehicle eliminates concerns about interfering with production schedules.
- Scalability of insights – The smaller vehicle is a scaled-down version of the CMTI platform, meaning that successful simulation results should translate well to full-scale implementation.
- Easier parameter adjustments – Variables such as the centre of mass location can be easily modified in a small vehicle, whereas adjusting the CMTI platform's centre of mass would be significantly more complex.

We therefore prescribe the following roadmap as a continuation of this work:

1. Develop and refine the simulation digital twin for an existing small skid-steer robot, ensuring accurate replication of vehicle dynamics and terrain interactions.
2. Calibrate and validate the model using experimental data from the physical test vehicle, refining the simulation as needed.
3. Once the small-scale simulation is sufficiently accurate, adapt and expand the model to replicate the full CMTI platform, leveraging existing data from this project for calibration.
4. Conduct and compare underground TRL6 tests with the validated CMTI simulation twin.

This structured approach represents a low-risk, high-impact strategy for advancing simulation-driven mining technology. By validating the digital twin at a small scale before full-scale implementation, we ensure that South African mining operations can significantly enhance efficiency and safety with minimal input costs. Once the methodology is proven, extending the simulation twin concept to other mining vehicles should be straightforward, offering a scalable and adaptable solution for the industry.

8 REFERENCES

Harrison, G. (2008). Anglo Platinum Extra Low Profile (XLP) Mechanised Equipment Implementation–An Update. *The Southern African Institute of Mining and Metallurgy*, *108*, 293-296.

Jordaan, J. (2003). Bord-and-pillar mining in inclined orebodies. *Journal of the Southern African Institute of Mining and Metallurgy*, *103*(2), 101-110.

Labbe, R. (2014). Kalman and Bayesian filters in Python. *Chap*, *7*(246), 4. https://github.com/rlabbe/Kalman-and-Bayesian-Filters-in-Python.

van den Berg, T. (2000). *Mechanised mining at a semi-steep dip* Association of Mine Managers South Africa,

van Eyk, L., & Heyns, P. S. (2024). A framework to define, design and construct digital twins in the mining industry. *Computers & Industrial Engineering*, 110805. https://doi.org/10.1016/j.cie.2024.110805.