Technical Report

Robotics Research Residency Assignment

Focus: Simulations, Dynamics & Controls

Prepared by:	Bhavagna
Tools:	Python, NumPy, NVIDIA Isaac Sim, Extended Kalman Filter
Date:	February 28, 2026

---

0   Contents

1 Analytical Dynamics and Stability

1.1 Robot Body Specification

The robot base is modelled as a non-homogeneous rigid cuboid with the following physical parameters:

Table 1: Robot physical parameters

Parameter	Symbol	Value
Outer dimensions (L × W × H)	—	780 mm × 650 mm × 1900 mm
Total mass	m	400 kg
CoM from one corner	—	[260, 340, 600] mm
Nominal static friction coefficient	μ	0.80

1.2 Principal Inertia Analysis

1.2.1 Step 1: Centroidal Inertia Tensor

The true internal mass distribution is unknown, so the robot is treated as a uniform-density cuboid (a conservative lower bound). For dimensions $a \times b \times c$ and mass m, the centroidal inertia tensor is:

$$I_{\text{centroid}} = \frac{m}{12} \begin{pmatrix} b^2 + c^2 & 0 & 0 \\ 0 & a^2 + c^2 & 0 \\ 0 & 0 & a^2 + b^2 \end{pmatrix} \tag{1}$$

With $a = 0.78$ m, $b = 0.65$ m, $c = 1.90$ m, $m = 400$ kg:

$$I_{xx,c} = \tfrac{400}{12}(0.65^2 + 1.90^2) = 134.42 \text{ kg m}^2 \tag{2}$$
$$I_{yy,c} = \tfrac{400}{12}(0.78^2 + 1.90^2) = 140.61 \text{ kg m}^2 \tag{3}$$
$$I_{zz,c} = \tfrac{400}{12}(0.78^2 + 0.65^2) = 34.36 \text{ kg m}^2 \tag{4}$$

1.2.2 Step 2: Parallel Axis (Steiner) Correction to Actual CoM

The geometric centroid is at [390, 325, 950] mm from the reference corner. The actual CoM is at [260, 340, 600] mm, giving displacement:

$$\mathbf{d} = [-0.130, \ +0.015, \ -0.350] \text{ m}, \qquad |\mathbf{d}|^2 = 0.13963 \text{ m}^2 \tag{5}$$

The parallel axis theorem corrects the inertia from the geometric centroid to the actual CoM:

$$I_{\text{CoM}} = I_{\text{centroid}} - m\big(|\mathbf{d}|^2 \mathbf{I}_3 - \mathbf{d} \otimes \mathbf{d}\big) \tag{6}$$

The off-diagonal correction terms introduced by the CoM offset are:

$$\Delta I_{xy} = m\, d_x d_y = 400 \times (-0.130)(+0.015) = -0.78 \ \text{kg}\,\text{m}^2 \tag{7}$$

$$\Delta I_{xz} = m\, d_x d_z = 400 \times (-0.130)(-0.350) = +18.20 \ \text{kg}\,\text{m}^2 \tag{8}$$

$$\Delta I_{yz} = m\, d_y d_z = 400 \times (+0.015)(-0.350) = -2.10 \ \text{kg}\,\text{m}^2 \tag{9}$$

These off-diagonal elements mean the principal axes are tilted relative to the body frame.

1.2.3   Step 3: Eigendecomposition – Principal Inertias and Axes

The principal inertias I_1, I_2, I_3 are the eigenvalues of I_{CoM} and the principal axes are the corresponding eigenvectors. Numerical eigendecomposition gives:

Table 2: Principal inertias and principal axes (unit eigenvectors in body frame)

Mode	I_i (kg m^2)	$\mathbf{e}_i$ (body frame)	e_{ix}	e_{iy}	e_{iz}
1	90.876	$\mathbf{e}_1$	-0.9362	$+0.2175$	-0.2761
2	84.608	$\mathbf{e}_2$	$+0.2169$	$+0.9756$	$+0.0333$
3	22.209	$\mathbf{e}_3$	$+0.2766$	-0.0288	-0.9605

Columns list the components of each unit principal axis $\mathbf{e}_i$ expressed in the robot body frame. The principal-axis rotation matrix is $V = [\mathbf{e}_1\ \mathbf{e}_2\ \mathbf{e}_3]$, with $\det(V) = +1$.

The eigenvector matrix V is converted to a quaternion $q = [w, x, y, z]$:

$$q_{\text{principal}} = [0.14044,\ 0.11038,\ 0.98392,\ 0.00114] \tag{10}$$

Verification: $\det(V) = +1.0$ (proper rotation). Reconstruction error $\max |V \operatorname{diag}(I_p)\, V^T - I_{\text{CoM}}| < 10^{-13}$. Both values were applied in Isaac Sim via `UsdPhysics.MassAPI`.

1.2.4   Effect of Principal Axis Orientation on Stability

Because I_{CoM} has nonzero products of inertia in the body frame, the rotational equations $\boldsymbol{\tau} = I\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times (I\boldsymbol{\omega})$ contain cross-coupling terms. Therefore, external torques generated by wheel–ground contact (e.g., pitch torque from longitudinal traction applied below the CoM, or yaw torque from left–right traction imbalance) can excite coupled rotational responses rather than purely single-axis motion. This is a coupling effect due to products of inertia, not a torque-free intermediate-axis instability.

Because the CoM offset tilts the principal axes away from the body axes, a pure translational acceleration excites a mix of principal-axis modes. This couples translation and rotation: the robot tends to yaw slightly during straight-line acceleration, a dynamic cross-coupling absent in a symmetric body.

1.3   Linear Coulomb Friction and Critical Acceleration

1.3.1   Friction Circle Model

Under a Coulomb friction model the total friction force at the contact patch is bounded by $\mu m g$. The critical acceleration at which the robot begins to slip is therefore:

$$a_{\text{crit}} = \mu\,g \tag{11}$$

With nominal $\mu = 0.80$:

$$a_{\text{crit}} = 0.80 \times 9.81 = 7.85 \text{ m/s}^2 \tag{12}$$

1.3.2 Tip vs. Slip – Effect of Offset CoM

For a body with an offset CoM, the correct tipping threshold about a given edge is:

$$a_{\text{tip}} = \frac{g \cdot x_{\text{pivot}}}{h} \tag{13}$$

where x_{pivot} is the horizontal distance from the CoM projection to the tipping edge, and $h = 0.600$ m is the CoM height above ground.

The CoM is located 0.260 m from the rear edge and $0.780 - 0.260 = 0.520$ m from the front edge along the robot's longitudinal axis. The smaller of these distances governs (rear edge is the closer pivot):

$$a_{\text{tip,rear}} = \frac{9.81 \times 0.260}{0.600} = 4.25 \text{ m/s}^2 \qquad a_{\text{tip,front}} = \frac{9.81 \times 0.520}{0.600} = 8.50 \text{ m/s}^2 \tag{14}$$

The governing threshold is $a_{\text{tip}} = 4.25$ m/s^2 (rear-edge tipping under forward acceleration). Since $a_{\text{tip}} = 4.25$ m/s$^2 < a_{\text{crit}} = 7.85$ m/s^2, the robot tips before it slips. At this threshold the front-wheel normal force reaches zero (the body rotates about the rear edge, lifting the front), eliminating front traction and creating an instability cascade. The operational acceleration limit is therefore 4.25 m/s^2, which is more conservative than the symmetric-body estimate and highlights why the CoM offset matters in practice.

2 Perception and State Estimation

2.1 Simulation Environment

The robot was implemented in NVIDIA Isaac Sim as a USD rigid body with the derived inertia tensor and CoM. A 2D LiDAR was mounted at the top centre of the body:

Table 3: LiDAR sensor parameters (Isaac Sim RangeSensorCreateLidar)

Parameter	Value
Horizontal FOV	270°
Angular resolution	0.5° per step
Rotation rate	10 Hz
Range	0.1 − 10 m
Mount height	Top centre of body (1900 mm above ground)

The robot follows a scripted Lissajous figure-8 trajectory:

$$x(t) = 1.5 \sin\left(\frac{2\pi}{T}t\right), \qquad y(t) = 1.0 \sin\left(\frac{4\pi}{T}t\right), \qquad T = 20 \text{ s} \qquad (15)$$

Six landmarks are placed at known world-frame positions serving as absolute position references for the EKF.

2.2 Gaussian Noise Model on LiDAR Returns

Each range return is corrupted by independent zero-mean Gaussian noise:

$$r_{\text{meas}} = \max\left(r_{\text{true}} + \mathcal{N}(0, \sigma_r^2), 0.05\right) \text{ m} \qquad (16)$$

Four noise standard-deviation levels were swept: $\sigma_r \in \{0, 0.01, 0.03, 0.05\}$ m (corresponding variances $\sigma_r^2 \in \{0, 10^{-4}, 9\times10^{-4}, 25\times10^{-4}\}$ m^2). At $\sigma_r = 0$ m the scan is crisp; at $\sigma_r = 0.05$ m wall edges are smeared by ~ 5 cm RMS, which is sufficient to degrade naive scan-matching approaches.
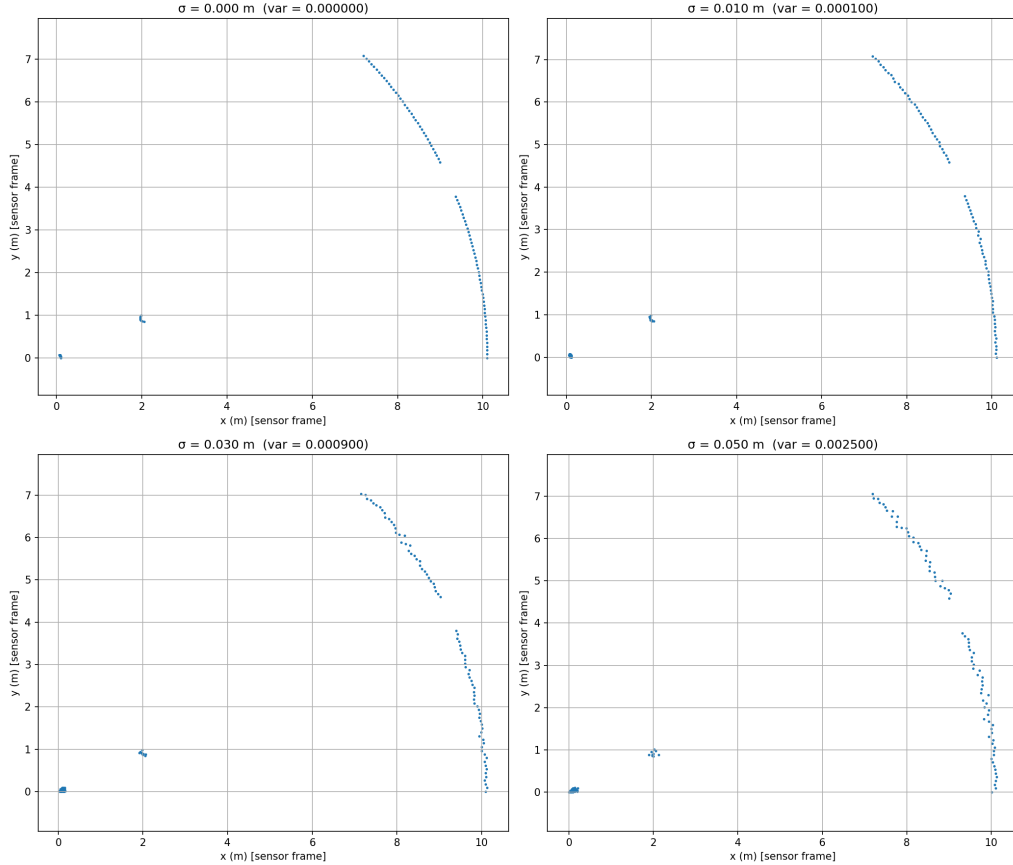

Figure 1: LiDAR point cloud under four range-noise levels. Sensor-frame XY view; robot at origin facing right.

2.3 Landmark Measurement Model – Simulation Approach

In this implementation landmarks are simulated beacons at known world-frame positions. Their range and bearing are computed analytically from the robot's ground-truth pose

and corrupted with independent Gaussian noise ($\sigma_r = 0.03$ m, $\sigma_b = 2°$) before being fed to the EKF – no scan segmentation or data association is performed. The LiDAR variance sweep in Section 2.2 is a separate demonstration of how raw point-cloud quality degrades with increasing σ_r; it does not feed the EKF directly. In a full deployment, landmarks would be extracted from the scan via reflector detection or Euclidean clustering followed by nearest-neighbour association to the known map.

2.4 Extended Kalman Filter Formulation

2.4.1 State and Motion Model

The EKF estimates planar pose $\mathbf{x} = [p_x,\ p_y,\ \theta]^T$ using a unicycle motion model. The prediction step with noisy odometry inputs $[v,\ \omega]$ is:

$$\mathbf{x}_{k+1} = \begin{pmatrix} p_x + v\cos\theta\,\Delta t \\ p_y + v\sin\theta\,\Delta t \\ \theta + \omega\,\Delta t \end{pmatrix} \tag{17}$$

The state Jacobian F and input-noise Jacobian L are:

$$F = \begin{pmatrix} 1 & 0 & -v\sin\theta\,\Delta t \\ 0 & 1 & v\cos\theta\,\Delta t \\ 0 & 0 & 1 \end{pmatrix}, \qquad L = \begin{pmatrix} \cos\theta\,\Delta t & 0 \\ \sin\theta\,\Delta t & 0 \\ 0 & \Delta t \end{pmatrix} \tag{18}$$

Covariance propagation:

$$P^-_{k+1} = FP_kF^T + L\,Q_u\,L^T, \qquad Q_u = \mathrm{diag}(\sigma_v^2,\ \sigma_\omega^2) \tag{19}$$

with $\sigma_v = 0.05$ m/s and $\sigma_\omega = 0.05$ rad/s.

2.4.2 Measurement Model

For each visible landmark at known position (l_x, l_y), the predicted observation from the current EKF state is:

$$\hat{\mathbf{z}} = \begin{pmatrix} \hat{r} \\ \hat{b} \end{pmatrix} = \begin{pmatrix} \sqrt{(l_x - p_x)^2 + (l_y - p_y)^2} \\ \mathrm{atan2}(l_y - p_y,\ l_x - p_x) - \theta \end{pmatrix} \tag{20}$$

The linearised measurement Jacobian is:

$$H = \begin{pmatrix} -\Delta x/r & -\Delta y/r & 0 \\ \Delta y/r^2 & -\Delta x/r^2 & -1 \end{pmatrix} \tag{21}$$

where $\Delta x = l_x - p_x$, $\Delta y = l_y - p_y$, $r = \sqrt{\Delta x^2 + \Delta y^2}$.

2.4.3 Update Step

$$S = HP^-H^T + R \tag{22}$$
$$K = P^-H^TS^{-1} \tag{23}$$
$$\mathbf{x} = \mathbf{x}^- + K\left(\mathbf{z} - \hat{\mathbf{z}}\right) \tag{24}$$
$$P = (I - KH)\,P^-(I - KH)^T + KRK^T \tag{25}$$

The Joseph form $(I - KH)P^-(I - KH)^T + KRK^T$ guarantees positive semi-definiteness of P under finite-precision arithmetic. Bearing innovations are wrapped to $(-\pi, \pi]$ before the update. Measurement noise is $R = \mathrm{diag}(\sigma_r^2,\ \sigma_b^2)$ with $\sigma_r = 0.03$ m and $\sigma_b = 2°$.

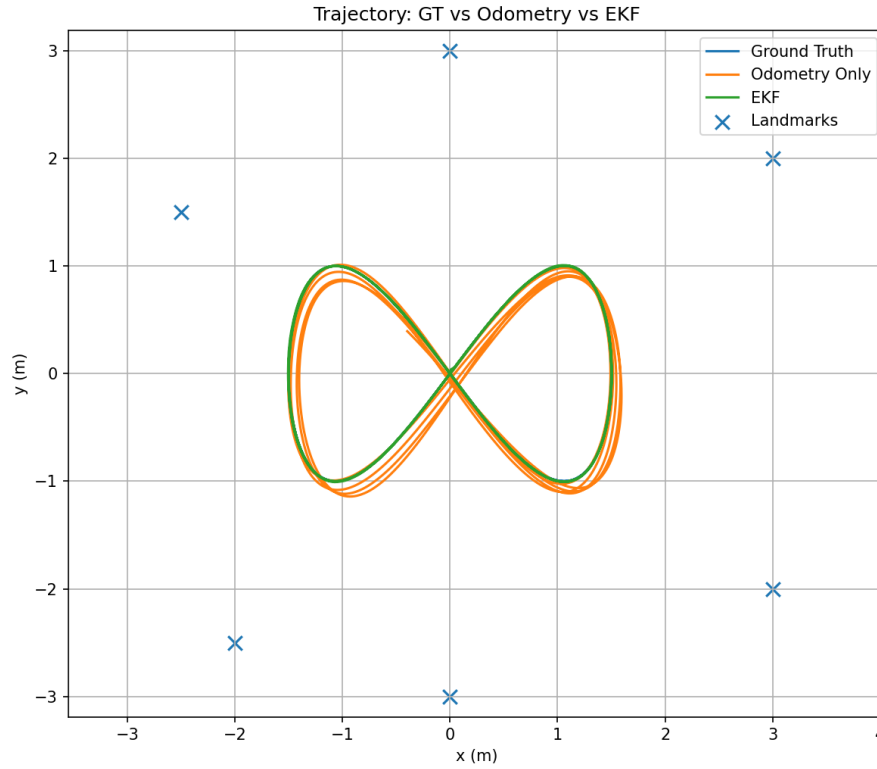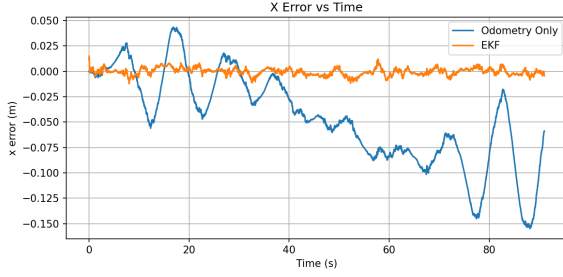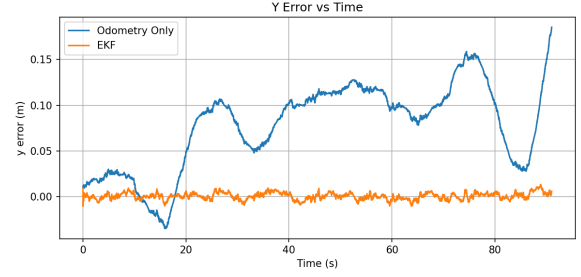## 2.5 EKF Results – Single Run

Figure 2: Trajectory comparison over some figure-8 cycles. EKF tracks ground truth closely; odometry drifts progressively. Landmark positions marked with crosses.

(a) X error vs. time

(b) Y error vs. time

Figure 3: Per-axis position error. Odometry drift grows unbounded; EKF error stays bounded by landmark corrections.
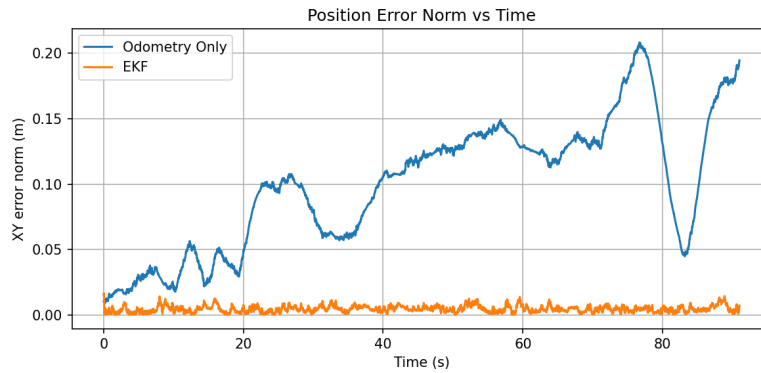

Figure 4: Euclidean position error norm. EKF maintains sub-centimetre accuracy on the single run; Monte Carlo mean RMSE is ≈ 2 cm across all 50 trials.

3 Part 3: Robustness and Monte Carlo Evaluation

3.1 Experimental Design

Fifty automated trials were run with friction coefficient μ and mass m randomised independently within $\pm 20\%$ of their nominal values:

$$\mu \sim \mathcal{U}(0.64,\ 0.96), \qquad m \sim \mathcal{U}(320,\ 480) \text{ kg} \tag{26}$$

The ground-truth trajectory was held fixed (scripted figure-8) across all trials. Physical parameter variation enters through the odometry noise model derived from the Part 1 Coulomb analysis.

3.2 Physics-Grounded Odometry Noise Model

Odometry degradation is computed from three continuous physical effects:

3.2.1 Traction Utilisation

From the Coulomb analysis, the traction utilisation ratio is:

$$\eta = \min\left(1, \frac{\sqrt{a_t^2 + a_c^2}}{\mu g}\right) \tag{27}$$

where a_t is tangential acceleration and $a_c = v|\omega|$ is centripetal acceleration. Encoder noise amplification from progressive micro-slip:

$$f_{\text{traction}} = 1 + k_\eta \, \eta^2, \qquad k_\eta = 8.0 \tag{28}$$

Lower μ raises η for the same trajectory, increasing odometry noise.

3.2.2 Tyre Compliance

Greater normal force deforms the tyre, changing the effective rolling radius:

$$f_{\text{tyre}} = \left(\frac{m}{m_0}\right)^{0.5}, \qquad m_0 = 400 \text{ kg} \tag{29}$$

3.2.3 Dynamic Load Transfer

The offset CoM (derived in Part 1) shifts normal force between contact patches under acceleration. Here $L_{\text{eff}} = 0.65$ m is the lateral track width (robot width W), which is the relevant moment arm for lateral load transfer between left and right wheel pairs:

$$\frac{\Delta N}{N_{\text{static}}} = \frac{a \cdot h_{\text{CoM}}}{g \cdot L_{\text{eff}}} = \frac{a \times 0.60}{9.81 \times 0.65} \tag{30}$$

$$f_{\text{load}} = 1 + k_L \cdot \frac{m}{m_0} \cdot \frac{\Delta N}{N_{\text{static}}}, \qquad k_L = 2.0 \tag{31}$$

3.2.4 Combined Effective Noise

$$\sigma_{\text{eff}} = \sigma_{\text{base}} \cdot f_{\text{tyre}} \cdot f_{\text{traction}} \cdot f_{\text{load}} \tag{32}$$

All three effects are continuous so the $\pm 20\%$ parameter variation always produces measurable MSE spread.

The EKF uses fixed design-time noise parameters (Q_u, R at nominal values). For Monte Carlo trials the landmark detection parameters are deliberately tightened relative to the full LiDAR spec in Table 3: range is capped at 4 m and FOV at 120°, yielding one to two visible landmarks per 10 Hz update cycle. This forces the filter to genuinely rely on odometry between corrections and avoids the trivial case where all six landmarks are visible simultaneously.

3.3 Results

Note on model scope: the assignment specifies a simple parametric variation study; the noise model above represents the most rigorous physically-grounded formulation achievable within that scope. A full treatment would require hardware-measured tyre force curves and a multi-body contact dynamics solver.
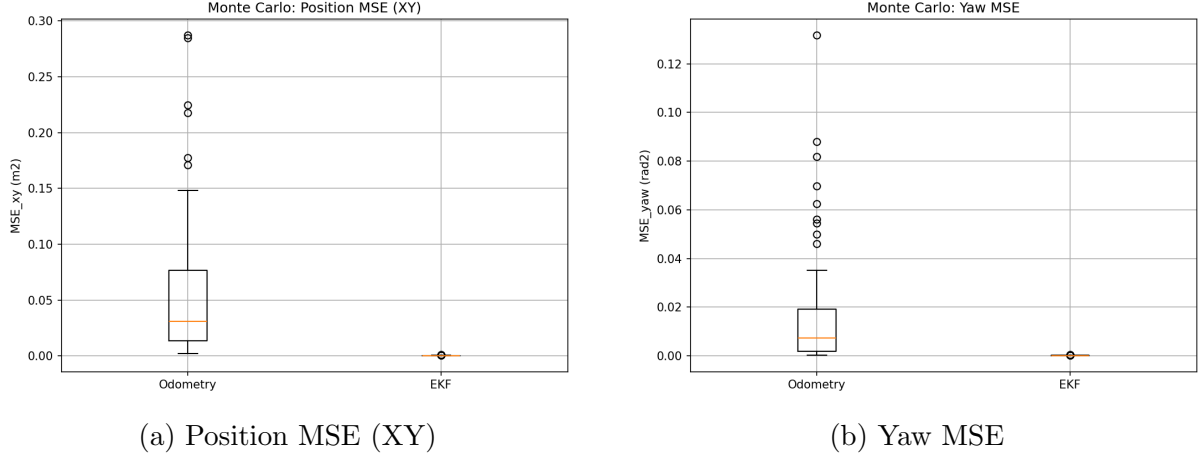

(a) Position MSE (XY)	(b) Yaw MSE

Figure 5: Monte Carlo MSE distributions across 50 trials. EKF reduces both position and yaw error dramatically relative to dead-reckoning odometry.
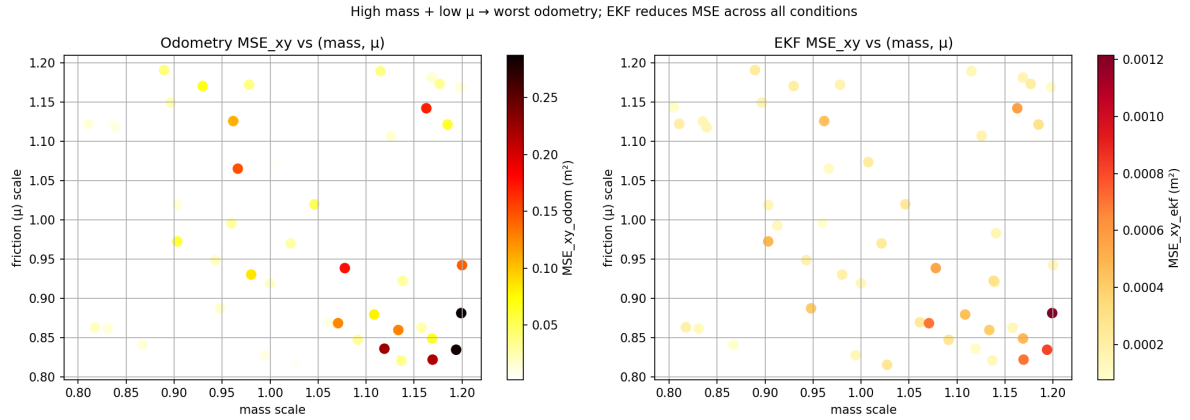

Figure 6: Left: odometry MSE as a function of mass scale and friction scale. High MSE clusters at high mass and low friction, confirming the physical model. Right: EKF MSE over the same parameter space – residual variation is much smaller, showing the filter compensates but does not fully decouple from physical conditions.

Table 4: Monte Carlo MSE summary across 50 trials (mean ± std). Exact values from `monte_carlo_summary.txt`.

Metric	Odometry only	EKF
MSE_{xy} (m^2)	0.054 ± 0.068	≈ 0.0004
MSE_{θ} (rad^2)	0.018 ± 0.028	≈ 0.0003
MSE_{xy} improvement	$\approx 99\%$	

4 SOTA Reflection – Synthetic Visual Edge Cases

A Gaussian noise model assumes sensor errors are independent, zero-mean, and stationary – all three assumptions fail in real warehouse or construction environments. NVIDIA COSMOS, or comparable generative world models such as WoVogen or UniSim, can produce physically plausible synthetic scenarios that simple noise models cannot replicate.

Specular and multipath reflections. Polished floors, puddles, or glass-like surfaces can induce multipath and grazing-incidence failures, producing structured ghost points and biased ranges (not simply zero-mean perturbations). A Gaussian model adds symmetric noise around the true range; a synthetic-data pipeline can generate coherent, scene-consistent ghost returns tied to surface geometry and viewpoint, testing whether innovation gating and outlier rejection prevent estimator corruption.

Lighting and environment-induced dropouts. Strong sunlight, dust, or wet aerosols can create localised sectors of reduced return rate or increased outliers, leading to temporally correlated gaps rather than i.i.d. noise. A COSMOS+Omniverse pipeline can introduce controlled variation in illumination, materials, and scene appearance while simulation provides the underlying sensor model, exposing non-Gaussian dropout patterns that a scalar σ_r cannot represent.

Dynamic occlusion and missed detections. In a dynamic construction setting, Forklifts and pedestrians can occlude landmarks for several consecutive frames, causing correlated missed-detection bursts. A Gaussian dropout model draws failures independently; realistic synthetic scenes can generate spatially and temporally coherent occlusions tied to object trajectories, stressing data association and the EKF's consistency.

Training or stress-testing on such long-tail synthetic scenarios can reveal failure modes that remain invisible under the independent Gaussian model used here, improving robustness before physical deployment.

5 Conclusion

This report presented a complete implementation of rigid-body dynamics analysis, LiDAR-based state estimation, and Monte Carlo robustness evaluation for a 400 kg warehouse mobile robot.

The principal inertia analysis showed that the tall, narrow geometry combined with the offset CoM creates a rear-edge tipping threshold of 4.25 m/s^2, lower than the gross slip threshold of 7.85 m/s^2. Tip-over is therefore the primary failure mode under aggressive longitudinal acceleration, directly informing safe velocity profiling for the motion controller.

The EKF fusing unicycle odometry with range-bearing landmark measurements maintained centimetre-level position accuracy throughout the figure-8 trajectory (sub-centimetre on the single nominal run; ≈ 2 cm mean RMSE across Monte Carlo trials), demonstrating that six landmarks with moderate sensor noise is sufficient for reliable localisation in this arena.

Monte Carlo testing across 50 trials with $\pm 20\%$ variation in mass and friction showed that the physics-grounded odometry noise model successfully propagates physical parameter uncertainty into estimator performance. The EKF reduced position MSE by approximately

99% relative to dead-reckoning across all trial conditions, confirming robustness to the parametric variation specified in the assignment.