Real-Time Humanoid State Estimation with IMU, Kinematic Odometry, and Visual SLAM

William Burgin and **Thomas O'Brien**
University of Newcastle, Australia
william.burgin@uon.edu.au, thomas.obrien@uon.edu.au

Abstract

We present a real-time humanoid state estimation method that loosely fuses inertial attitude (Mahony filter), kinematic odometry from legged locomotion, and a visual SLAM frontend based on the open-source Stella system. Humanoid locomotion induces rapid viewpoint changes, motion blur, and contact-driven dynamics that challenge conventional visual and visual-inertial pipelines. Building on a decade of humanoid-focused estimation research, we design a lightweight fusion architecture in which the IMU provides high-rate roll and pitch, kinematics anchor vertical position and short-term translation, and vision supplies drift-reducing corrections in translation and heading. A sliding-window smoother further enforces temporal consistency. We compare three configurations in real-time on a humanoid platform: (i) IMU plus kinematic odometry, (ii) visual SLAM alone, and (iii) the fused system. Experiments evaluate accuracy and robustness and conclude that the fused approach achieves lower drift than either component alone while maintaining real-time performance.

1 Introduction

Accurate onboard state estimation is essential for humanoids to walk, navigate, and interact safely. Unlike wheeled robots, they lack wheel odometry and experience contact-driven dynamics with periodic accelerations [Oriolo *et al.*, 2016]. Visual SLAM offers rich exteroceptive information but fails under abrupt motion, blur, or low texture [Stasse *et al.*, 2006; Tsotsos *et al.*, 2012; Scona *et al.*, 2017; Mur-Artal and Tardós, 2017]. IMUs provide high-rate rotation but drift when integrated [Mahony *et al.*, 2008]. Kinematic odometry stabilises short-term motion but suffers from compliance and foot slippage [Oriolo *et al.*, 2012; Ahn *et al.*, 2012; Fallon *et al.*, 2014].

This work addresses these limitations by combining the complementary strengths of all three modalities. Our approach is a loosely coupled design: a Mahony filter provides drift-resilient attitude, kinematic odometry supplies body translation through anchor point kinematics, and a visual SLAM frontend delivers global corrections. To improve consistency, a sliding-window optimizer refines recent body poses by balancing smoothness with agreement to both visual and kinematic measurements. The design emphasises computational efficiency and real-time operation on humanoid platforms rather than tightly coupled optimization.

The contributions of this paper are: (i) a practical, real-time estimation architecture that blends IMU attitude, kinematic dead reckoning, and visual pose updates; (ii) a lightweight online method for resolving the scale ambiguity of monocular SLAM using ground-plane geometry; and (iii) experimental comparisons between IMU plus kinematics, visual SLAM alone, and the fused system on a humanoid robot.

2 Related Work

Early real-time monocular and stereo SLAM demonstrations on humanoids showed feasibility but also highlighted gait-induced challenges. Stasse et al. adapted monocular EKF-SLAM to a humanoid and leveraged pattern generator information to stabilize tracking [Stasse *et al.*, 2006]. Kwak et al. combined stereo vision with particle-filter SLAM to handle feature ambiguity, mapping while localizing [Kwak *et al.*, 2009]. These efforts demonstrated real-time operation but also exposed sensitivity to rapid, periodic motion.

Subsequent approaches explicitly addressed gait effects and aggressive motion. Tsotsos et al. proposed a visual-inertial odometry tailored for humanoids with multi-scale feature tracking and a kinematic-statistical motion model to capture quasi-periodic dynamics [Tsotsos *et al.*, 2012]. Ahn et al. fused leg kinematic odometry, IMU, and stereo visual odometry within an EKF to provide a stabilized motion prior to vision, improving ro-

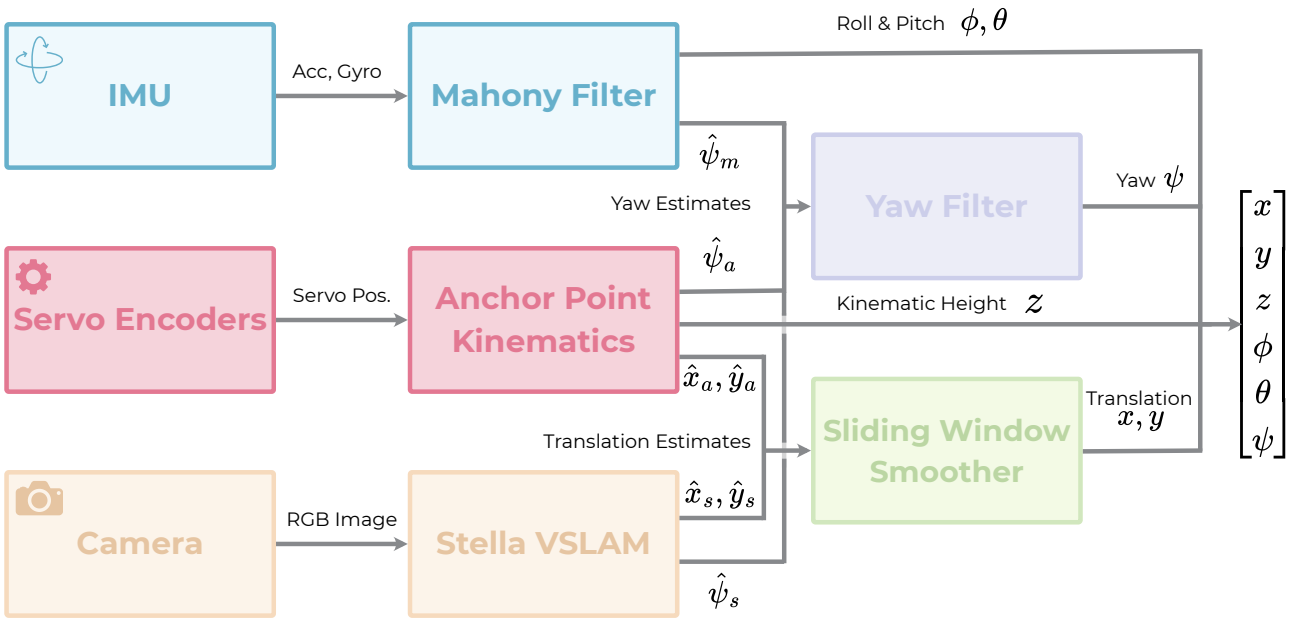


Figure 1: Humanoid State Estimation Scheme

bustness under fast walking [Ahn *et al.*, 2012]. Oriolo et al. formalized kinematic-visual-inertial EKF localization with support-foot switching, enabling high-rate odometry prediction corrected by visual and inertial updates [Oriolo *et al.*, 2012; 2016].

Optimization-based pipelines further improved robustness by integrating priors from proprioception. Scona et al. demonstrated direct visual SLAM fused with proprioceptive priors on a humanoid, improving resilience to viewpoint changes, blur, and feature sparsity while maintaining real-time mapping [Scona *et al.*, 2017]. Related work in legged estimation fused lidar with IMU and kinematics for drift-free performance on Atlas, illustrating the value of exteroceptive loop closures in legged motion [Fallon *et al.*, 2014].

Modern visual SLAM frameworks such as ORB-SLAM2 and OpenVSLAM provide real-time feature-based tracking, mapping, and loop closure, and are widely used as frontends in robotics [Mur-Artal and Tardós, 2017; Sumikura *et al.*, 2019]. While not humanoid-specific, they are often combined with inertial and kinematic sensing in humanoid systems. In parallel, deep learning enhanced visual frontends emerged, including learned keypoints and matching as well as dense learned tracking (for example, learned features and end-to-end odometry). Although some learned systems approach real-time on GPU, many humanoid deployments still favor classical feature-based or direct methods due to onboard compute constraints.

While prior work demonstrates the benefit of combining visual, inertial, and kinematic sensing, most approaches rely on tightly coupled EKF or factor-graph formulations, which increase computational load and integration complexity on resource-limited humanoids. Furthermore, monocular pipelines typically lack online scale recovery suitable for dynamic walking, and few studies explicitly evaluate robustness under aggressive head motion or deliberate visual dropout. Our approach adopts a lightweight, loosely coupled design with online ground-plane scale estimation and adaptive fusion, enabling real-time operation and reliable recovery behaviour. This fills a practical gap between high-performance but computationally intensive frameworks and the needs of agile, real-world humanoid control.

Problem Statement

We consider a world-fixed inertial frame $\mathcal{W}$ and a body-fixed frame $\mathcal{B}$ rigidly attached to the humanoid robot midway between the hip yaw joints. The aim of this work is to provide a real-time estimate of the floating base pose, given by

$$\mathbf{H}_b^w = \begin{bmatrix} \mathbf{R}_b^w & \mathbf{r}_{B/W}^w \\ \mathbf{0} & 1 \end{bmatrix}, \tag{1}$$

where $\mathbf{R}_b^w \in SO(3)$ is the rotation from $\mathcal{B}$ to $\mathcal{W}$, and $\mathbf{r}_{B/W}^w \in \mathbb{R}^3$ is the body position in $\mathcal{W}$. Decomposing $\mathbf{R}_b^w$ into its Euler angle representation allows us to represent

(1) as vector

$$\begin{bmatrix} r^w_{B/W} \\ \Theta^w_b \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ \phi \\ \theta \\ \psi \end{bmatrix}, \tag{2}$$

where $\Theta^w_b = [\phi, \theta, \psi]^\top$ are roll, pitch, and yaw angles respectively. These six degrees-of-freedom form the basis of the experimental evaluation, with translation and orientation errors reported in RMS terms.

3 Our Approach

Our approach, visually depicted in Figure 1, targets real-time humanoid state estimation by combining three sensor inputs: IMU (gyroscope + accelerometer), servo position encoders and a monocular RGB camera. For each of the components of (2) we develop a pipeline best tailored for each degree-of-freedom.

3.1 Mahony Filter Attitude Estimation

We assume the humanoid robot is equipped with an inertial measurement unit (IMU), collocated with $\mathcal{B}$. The IMU contains gyroscopes, providing angular velocity measurements ω_b , and accelerometers, providing linear acceleration measurements a^b that include both gravity and linear accelerations caused by robot motions. We estimate the floating base orientation using the Mahony filter [Mahony *et al.*, 2008], a nonlinear complementary filter on $SO(3)$. It fuses gyroscope and accelerometer measurements with integral correction terms, and is governed by two gains K_p and K_i that control proportional and integral feedback. The final output of the Mahony filter is Euler angles

$$\hat{\mathbf{x}}_m = \left[\hat{\phi}_m, \hat{\theta}_m, \hat{\psi}_m\right]. \tag{3}$$

Limitations In the absence of a magnetometer, yaw cannot be directly inferred from gravity measurements. As a result, the Mahony filter produces a heading estimate that gradually drifts over time, while roll and pitch remain well-constrained and reliably observable.

3.2 Anchor Point Kinematic Odometry

Leveraging the servo position sensors, we employ the anchor point strategy [Caron *et al.*, 2019; O'Brien, 2024]: when a foot is in contact, an anchor point A on its sole is assumed fixed in $\mathcal{W}$ at $\mathbf{H}^w_a$. In the body frame $\mathcal{B}$, the anchor frame $\mathbf{H}^b_a$ is known from forward kinematics. This gives kinematics-based estimate of the floating base pose via

$$\mathbf{H}^w_b = \mathbf{H}^w_a \times (\mathbf{H}^b_a)^{-1}. \tag{4}$$

Each time a support foot switch occurs, detected from either relative foot z-height thresholds or force sensors, the anchor point is updated to the new support foot using relative kinematics between feet, illustrated in Figure 2. The final output of the anchor point strategy is pose

$$\hat{\mathbf{x}}_a = [\hat{x}_a, \hat{y}_a, \hat{z}_a, \hat{\phi}_a, \hat{\theta}_a, \hat{\psi}_a]. \tag{5}$$

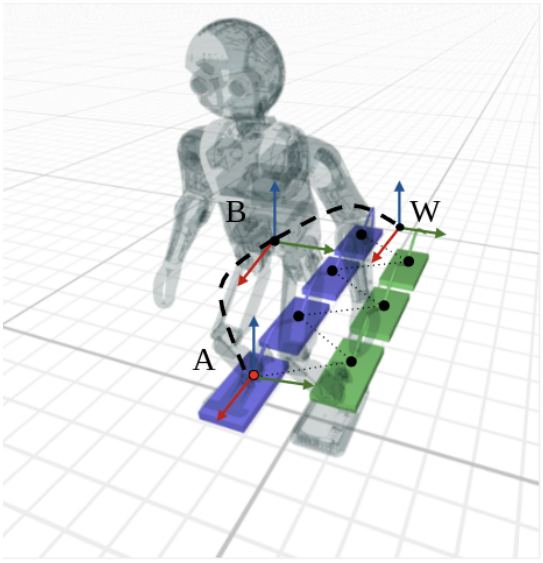
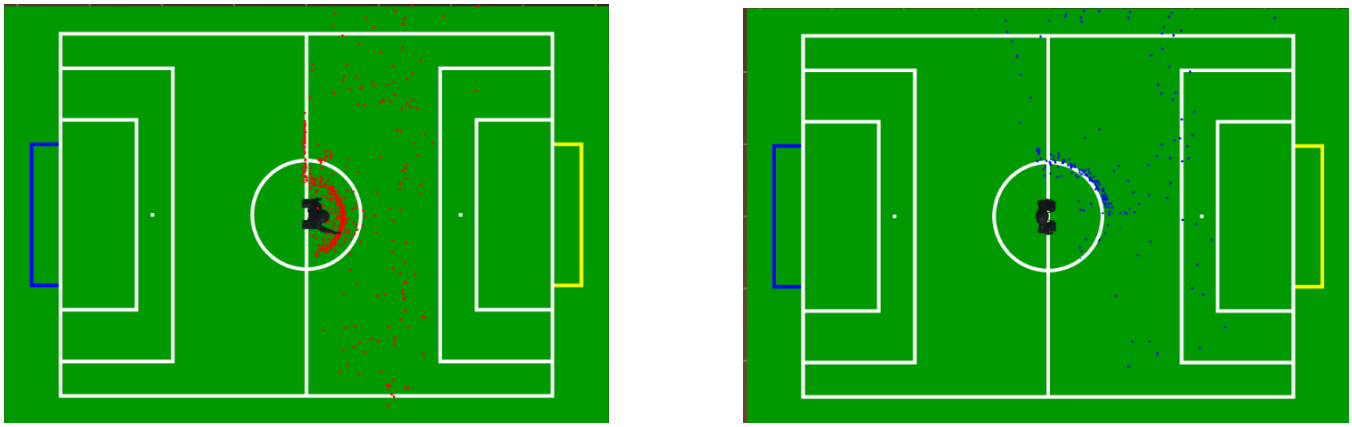


Figure 2: Kinematic odometry

Limitations While simple and computationally light, in practice, this anchor-point odometry accumulates noticeable drift in translation and orientation over time. Small violations of the fixed-contact assumption, e.g., slippage during stance, sole compliance, gearbox backlash, and encoder quantization, perturb the estimated anchor pose and are mapped through $(\mathbf{H}^b_a)^{-1}$ into the floating-base estimate.

3.3 Visual SLAM

For our visual SLAM implementation, we adopt the open source library Stella. Built as a fork of the OpenVSLAM [Sumikura *et al.*, 2019] framework, Stella combines a feature-based frontend, which detects and tracks ORB (Oriented FAST, Rotated Brief) keypoints, with a backend that performs keyframe-based pose graph optimisation. The backend also supports loop closure and relocalisation, producing a camera trajectory along with a sparse 3D map of the environment. At each update, Stella provides an unscaled camera pose $\tilde{\mathbf{H}}^n_c$ in its internal world frame $\mathcal{N}$. Upon startup, we compute via kinematics the transform $\mathbf{H}^w_n$ which maps this pose estimate into the common world frame $\mathcal{W}$. However, since



(a) Unscaled map points (red)

(b) Scaled map points (blue)

Figure 3: Comparison of Stella without and with scale correction.

we employ monocular SLAM, camera pose estimates $\tilde{\mathbf{H}}_c^n$ remain up to an arbitrary scale. We therefore introduce an online method to recover metric scale by exploiting ground-plane geometry.

Online Scale Estimation Monocular SLAM recovers structure only up to an unknown scale, which must be resolved to obtain metric state estimates. We address this by estimating a global scale factor s online through ground-plane intersections of map rays. This approach exploits the fact that during walking, the robot's body height relative to the support foot is well known from kinematics.

At each update, SLAM map points $\mathbf{r}_{P_i/N}^n$ are first transformed from the Stella frame $\mathcal{N}$ into the common world frame $\mathcal{W}$ via $\mathbf{H}_n^w$, yielding

$$\mathbf{r}_{P_i/W}^w = \mathbf{H}_n^w \, \mathbf{r}_{P_i/N}^n. \tag{6}$$

We define the camera–relative vector (expressed in $\mathcal{W}$)

$$\mathbf{r}_{P_i/C}^w = \mathbf{r}_{P_i/W}^w - \mathbf{r}_{C/W}^w. \tag{7}$$

Camera-relative rays are then formed from the camera position $\mathbf{r}_{C/W}^w$, with direction given by

$$\mathbf{u}_i = \frac{\mathbf{r}_{P_i/C}^w}{\|\mathbf{r}_{P_i/C}^w\|}. \tag{8}$$

Each ray is intersected with the ground plane $z = 0$, using the camera height $h = (\mathbf{r}_{C/W}^w)_z$. The intersection distance and ground-projected point are

$$d_i = \frac{|h|}{|(\mathbf{u}_i)_z|}, \qquad \mathbf{r}_{P_i/W,g}^w = \mathbf{r}_{C/W}^w + d_i \, \mathbf{u}_i. \tag{9}$$

We reject very shallow angled rays, to prevent numerically unstable and physically unreliable ground intersections, as well as rays whose ground intersections lie beyond 2 m from the camera. We also apply an image mask, part of the Stella module, that discards the upper 40% of each frame, as these regions mostly contain non-ground features. Future work should consider more principled methods for feature selection, such as leveraging ground-plane segmentation to filter out features that do not lie on the ground plane. For the remaining valid rays, the per-point scale is given by

$$s_i = \frac{\|\mathbf{r}_{P_i/W,g}^w - \mathbf{r}_{C/W}^w\|}{\|\mathbf{r}_{P_i/C}^w\|}, \tag{10}$$

and the overall scale estimate is taken as the average

$$\tilde{s}_k = \frac{1}{N} \sum_{i=1}^{N} s_i, \tag{11}$$

where N is the number of valid rays. To reduce jitter, the global scale factor is updated with exponential smoothing

$$s_k = \alpha_s s_{k-1} + (1 - \alpha_s)\tilde{s}, \tag{12}$$

Finally, the scaled estimate is applied to the translation component of the camera pose as

$$\mathbf{H}_c^n = \begin{bmatrix} \tilde{\mathbf{R}}_c^n & s_k \, \tilde{\mathbf{r}}_{C/N}^n \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix}, \tag{13}$$

and the reconstructed map points are rescaled accordingly, illustrated in Figure 3. This procedure allows metric scale to be recovered without additional sensors, while remaining lightweight enough for real-time operation. By constraining scale through repeated ground-plane intersections, we leverage the known body height from kinematics to resolve a fundamental limitation of

monocular SLAM. The combination of geometric filtering and exponential smoothing ensures the estimate is robust to noise and spurious features. The final output of Stella VSLAM component is body pose

$$\hat{\mathbf{x}}_s = [\hat{x}_s, \hat{y}_s, \hat{z}_s, \hat{\phi}_s, \hat{\theta}_s, \hat{\psi}_s]. \tag{14}$$

Limitations Despite loop closure, visual odometry inherently accumulates drift in both translation and orientation between loop events. In walking scenarios where the robot frequently changes viewpoint without revisiting the same location, accumulated drift can significantly degrade pose estimates. Furthermore, in scenarios where the camera is blocked by the robot's limbs, occluded by the environment, or operating in texture-poor areas (e.g., blank walls or low-light conditions), visual features may not be detected or tracked reliably. These dropouts can cause temporary loss of tracking or large jumps in pose when relocalisation occurs.

3.4 Filtering and Fusion

To address the limitations of each of the components detailed in Section 3.1, 3.2 & 3.3, we adopt a loosely coupled architecture in which each modality contributes complementary information.

Yaw filter

For orientation, the roll and pitch from the Mahony filter provide a reliable estimate, however, the yaw component gradually drifts over time. To address this, we introduce a lightweight complementary filter that combines gyroscope integration with kinematic yaw updates and visual SLAM corrections. The filter adaptively estimates and compensates for gyroscope bias while blending predictions and measurements in a complementary fashion. This approach stabilises yaw against long-term drift without incurring the computational overhead of full optimisation, making it well-suited for real-time humanoid operation. The implementation is available in our open source codebase on GitHub (see `https://github.com/NUbots/NUbots`), due to space constraints we omit detailed equations and implementation steps here.

Sliding-window smoothing

While vertical position (z) can be robustly obtained from kinematics, accurate horizontal translation (x, y) is more challenging. Anchor-point kinematic odometry accumulates drift due to foot slippage and compliance, while visual SLAM estimates are susceptible to short-term noise, dropouts, and scale uncertainty. To mitigate these issues, we employ a sliding–window smoothing approach that fuses both sources while enforcing temporal consistency. We further adaptively weight the contribution of Stella within the window based on its update status (e.g., tracking, relocalising, or lost), so that frames with confident visual updates influence the solution more strongly than uncertain ones.

We formulate a weighted least–squares problem over the most recent N body states. Let

$$\mathbf{x}_i = [x_i \;\; y_i]^\top, \qquad i = t - N + 1, \ldots, t, \tag{15}$$

denote the body (x, y) position in the world frame $\mathcal{W}$. The optimisation variable is the stacked vector

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_{t-N+1}^\top & \cdots & \mathbf{x}_t^\top \end{bmatrix}^\top \in \mathbb{R}^{2N}. \tag{16}$$

The cost balances smoothness with agreement to available measurements:

$$\begin{aligned} J(\mathbf{X}) \;=\; & \underbrace{\sum_{i=t-N+1}^{t-1} w_{sm} \left\| \mathbf{x}_{i+1} - \mathbf{x}_i \right\|^2}_{\text{smoothness}} \\ & + \underbrace{\sum_{i \in \mathcal{M}_s} w_{s,i} \left\| \hat{\mathbf{x}}_i^s - \mathbf{x}_i \right\|^2}_{\text{visual SLAM agreement (adaptive)}} \\ & + \underbrace{\sum_{i \in \mathcal{M}_a} w_{a,i} \left\| \hat{\mathbf{x}}_i^a - \mathbf{x}_i \right\|^2}_{\text{kinematic agreement}}. \end{aligned} \tag{17}$$

Here $\hat{\mathbf{x}}_i^s$ and $\hat{\mathbf{x}}_i^a$ denote measurements from Stella and kinematic odometry, $\mathcal{M}_s, \mathcal{M}_a$ are the index sets where such measurements are available, and w_{sm} is a fixed smoothness weight. The visual weights $w_{s,i}$ are time–varying and reflect the SLAM frontend status (e.g., high during stable tracking, reduced during relocalisation, and near-zero when lost); $w_{a,i}$ provides a weaker, steady prior from kinematics.

The optimisation is solved as

$$\mathbf{X}^\star = \arg\min_{\mathbf{X}} J(\mathbf{X}), \tag{18}$$

and the most recent state $\mathbf{x}_t^\star$ is taken as the current body translation estimate x and y. Earlier states in the window act as a buffer that propagates smoothness constraints. A moderate window size (e.g. $N = 10$–20) balances stability against computational cost, while larger values yield diminishing returns and increased latency.

4 Experiments

In this section we evaluate our state estimation approach onboard real humanoid hardware across four controlled scenarios, comparing IMU+kinematics, vision-only (Stella), and the fused system.

Table 1: RMS error comparison of Stella and IMU+Kinematics. Best results are highlighted in **bold**; ties are left unbolded.

Scenario	Method	X [m]	Y [m]	Z [m]	Roll [°]	Pitch [°]	Yaw [°]
Stella alone (with dropout)	Stella	0.479	**0.035**	0.004	52.55	51.16	7.63
	IMU+Kinematics	**0.070**	0.090	0.004	**3.22**	**15.35**	**6.21**
Pickup	Fused System	**0.063**	**0.032**	0.026	1.28	1.59	**1.48**
	IMU+Kinematics	N/A	N/A	0.026	1.28	1.59	4.08
Forward Walking	Fused System	**0.021**	**0.041**	0.004	1.27	1.75	**1.50**
	IMU+Kinematics	0.162	0.113	0.004	1.27	1.75	6.59
Visual Dropout	Fused System	**0.112**	**0.101**	0.005	1.43	2.21	**4.00**
	IMU+Kinematics	0.236	0.181	0.005	1.43	2.21	11.11

Table 2: RMS error comparison of Stella and IMU+Kinematics across experimental scenarios.

Scenario	Method	X [m]	Y [m]	Z [m]	Roll [°]	Pitch [°]	Yaw [°]
Stella alone (with dropout)	Stella	0.479	0.035	0.004	52.55	51.16	7.63
	IMU+Kinematics	0.070	0.090	0.004	3.22	15.35	6.21
Pickup	Fused System	0.063	0.032	0.026	1.28	1.59	1.48
	IMU+Kinematics	N/A	N/A	0.026	1.28	1.59	4.08
Forward Walking	Fused System	0.021	0.041	0.004	1.27	1.75	1.50
	IMU+Kinematics	0.162	0.113	0.004	1.27	1.75	6.59
Visual Dropout	Fused System	0.112	0.101	0.005	1.43	2.21	4.00
	IMU+Kinematics	0.236	0.181	0.005	1.43	2.21	11.11

Setup

The evaluation platform is the NUgus humanoid robot, developed by the NUbots team at the University of Newcastle. NUgus is a bipedal humanoid equipped with a monocular camera, IMU and joint encoders. Four representative scenarios were recorded: (i) a forward walking sequence, (ii) deliberate visual dropout with Stella running in isolation, (iii) a pickup test where the robot was carried to isolate gait-induced motion, and (iv) visual dropout where the lens was manually covered and uncovered. Ground truth was obtained from a motion capture system.

Metrics

Root-mean-square (RMS) errors in translation and orientation serve as the primary measure of accuracy (Table 2). Drift is assessed through accumulated errors in walking sequences, while robustness is evaluated by recovery following visual dropouts and deliberate occlusions. Qualitative assessment of trajectory smoothness and stability complements these metrics.

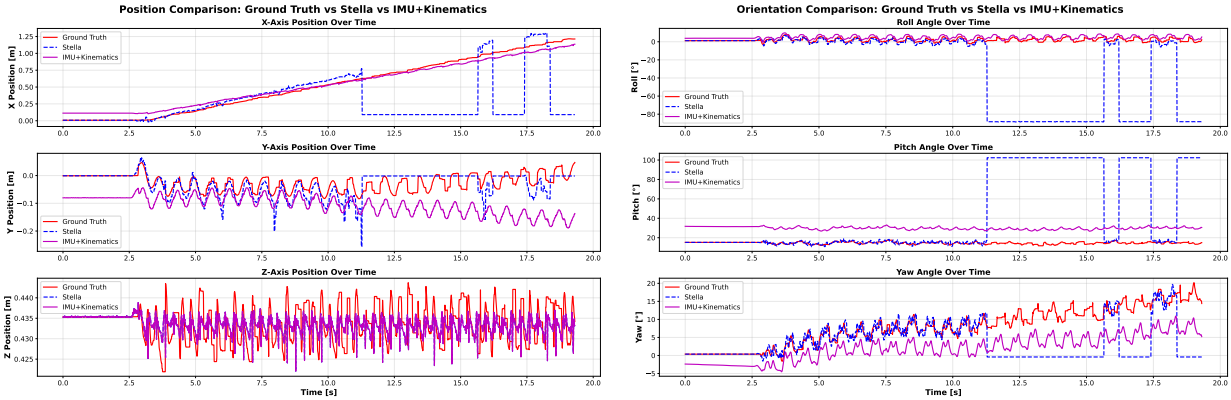
Baselines and Procedure

We compare three estimator configurations: (i) IMU plus kinematic odometry, where the Mahony filter provides attitude fused with kinematic translation, (ii) visual SLAM only, using the Stella frontend without IMU or kinematic input, and (iii) the fused system, which integrates Stella pose updates with Mahony attitude and kinematic odometry, and applies a sliding-window smoother for refinement. All runs use identical camera calibration and time-synchronized sensor streams.

4.1 Results and Discussion

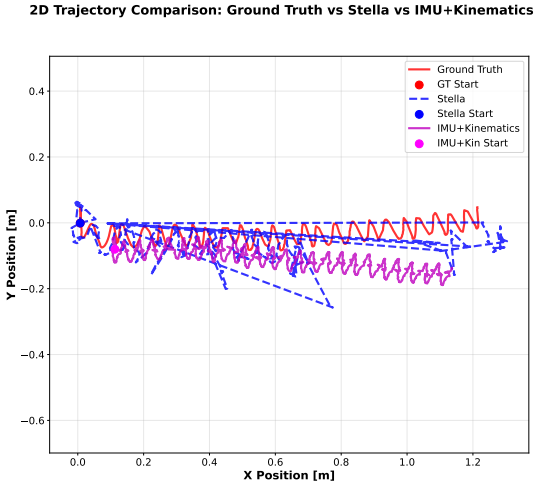
We evaluate four representative scenarios to illustrate how each system behaves under different conditions, and to highlight the complementary strengths of vision and kinematics. Each case is presented with three plots: translation (x, y, z), orientation (ϕ, θ, ψ), and the xy trajectory.

With Stella acting in isolation, we highlight the brittleness of vision-only estimation under gait. As shown in Fig. 4, rapid head motion and texture-poor regions cause Stella to lose track, leading to collapse of the (x, y, ψ) estimate. RMS errors confirm the instability: roll and pitch drift beyond 50°, compared to 3°–15° for IMU+kinematics. This illustrates the weakness of vision-only approaches, whereas kinematics fails in the opposite way, with drift accumulating unchecked without an external reference.



(a) Translation (x, y, z)

(b) Rotation (ϕ, θ, ψ)



(c) xy trajectory

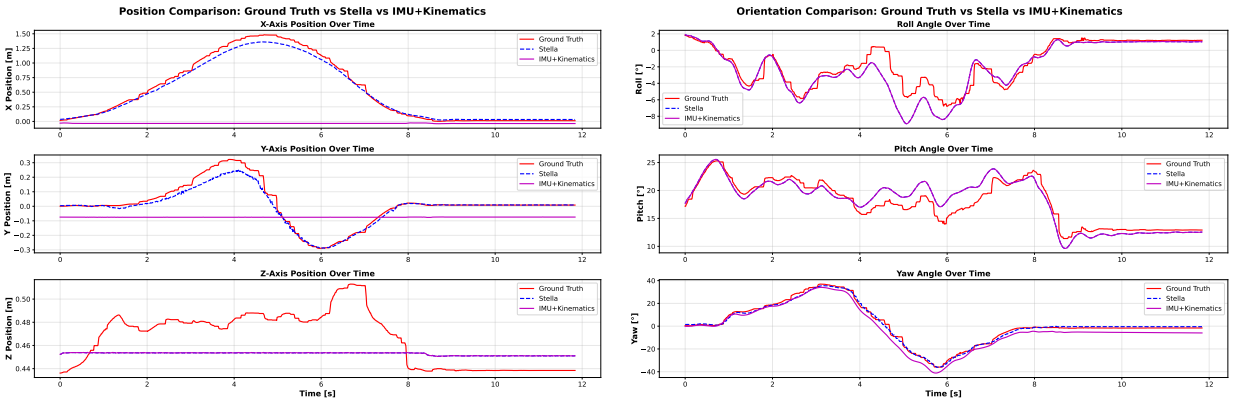
Figure 4: Stella alone, without kinematic or smoothing support.

To isolate the effect of gait-induced head oscillation, the robot was carried while running the fused system (Fig. 5). With no foot contacts, kinematics contribute little, yet Stella is able to deliver stable (x, y, ψ) estimates. Yaw error remains below 1.5°, whereas IMU+kinematics drifts to more than 4°. This demonstrates Stella's reliable yaw corrections in magnetometer-deprived environments. It also confirms that much of the visual noise seen during walking originates from the humanoid's head–neck design rather than limitations of the algorithm itself.

Building on this, the forward walking sequence (Fig. 6) outlines how vision anchors drift over extended motion. The fused system produces a smoother and more consistent trajectory than IMU+kinematics, which accumulates substantial error. Stella's long-term corrections substantially reduce drift, with translation RMS remaining under 5 cm in x and y, whereas IMU+kinematics drifts to 0.16 m and 0.11 m respectively. Yaw error is cut from 6.6° to 1.5°. This validates the central premise that kinematics provides short-term stability, but vision is needed to correct drift over time.

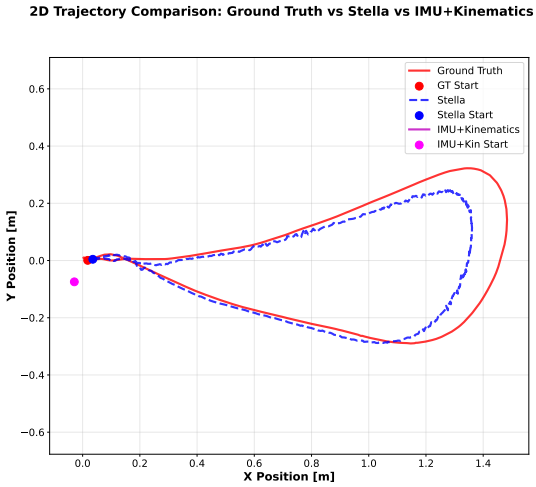
Finally, the visual dropout experiment (Fig. 7) stresses robustness. We note that this analysis assumes the presence of a reliable fault-detection module to flag visual failure, which is beyond the scope of this paper. With the lens deliberately covered (at 9.7 seconds), the fused system falls back to IMU and kinematics, avoiding catastrophic divergence. Once the camera is uncovered (at 15.6 seconds), Stella relocalises and the trajectory reconverges toward ground truth. Quantitatively, IMU+kinematics drifts beyond 0.2m in translation with yaw error above 12°, while Stella constrains drift to within 0.12m and 4° after recovery. This ability to degrade gracefully and recover when vision returns is critical for real-world deployment.

Across all scenarios, the fused estimator consistently outperforms individual modalities. Vision alone is brittle under gait-induced motion, while IMU+kinematics accumulates unbounded drift. Together, they provide complementary strengths. IMU and kinematics deliver stable short-term propagation, and Stella provides global



(a) Translation (x, y, z)

(b) Rotation (ϕ, θ, ψ)



(c) xy trajectory

Figure 5: Pickup experiment.

corrections in (x, y, ψ). The RMS analysis demonstrated yaw errors were reduced by a factor of 3–5 compared to IMU+kinematics, with significantly lower translational drift. Combined through online scale recovery and sliding-window smoothing, these components yield a robust estimator suitable for closed-loop humanoid control.

5 Conclusions

We have presented a real-time humanoid state estimator that combines IMU attitude, kinematic odometry, and a visual SLAM frontend. The design emphasises computational simplicity through a loosely coupled architecture: IMU attitude estimation provides roll, pitch, and a drift-prone yaw baseline; kinematic odometry anchors vertical position; and visual SLAM supplies drift-reducing corrections in horizontal translation and yaw. A sliding-window smoother further improves consistency. Experiments on a humanoid platform demonstrate that the fused system achieves lower drift and faster recovery from visual failures compared to either modality alone, while maintaining real-time performance.

Future work will explore tighter integration within a factor-graph framework, including IMU preintegration and foot-contact constraints, as well as learned visual frontends tailored to humanoid motion. These extensions aim to retain real-time operation while further improving robustness under aggressive gait dynamics.

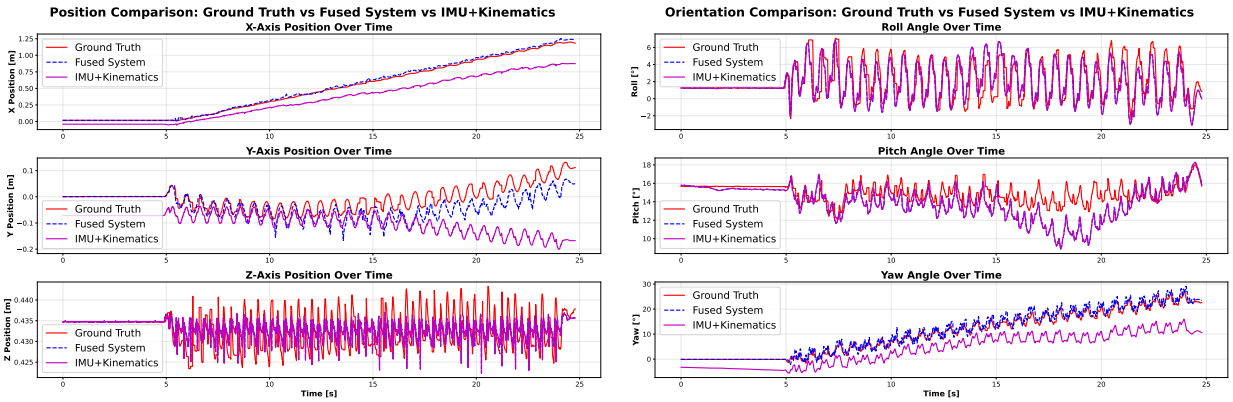
Acknowledgments

We thank colleagues and collaborators for discussions and support.

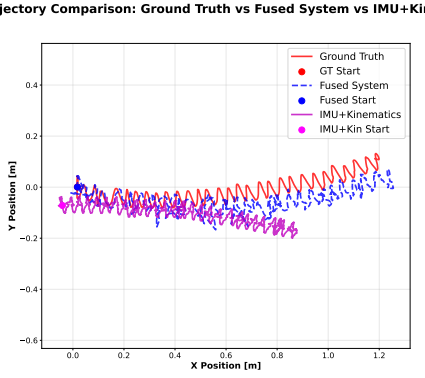
References

[Ahn *et al.*, 2012] Seungbae Ahn, Seonghyeon Yoon, Seokju Hyung, Nojun Kwak, and Kyung-Soo Roh. On-board odometry estimation for 3d vision-based slam of humanoid robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4006–4012, 2012.

[Caron *et al.*, 2019] Stéphane Caron, Abderrahmane Kheddar, and Olivier Tempier. Stair climbing sta-



(a) Translation (x, y, z) (b) Rotation (ϕ, θ, ψ)



(c) xy trajectory

Figure 6: Straight walking sequence with fused system.

bilization of the HRP-4 humanoid robot using whole-body admittance control. In *IEEE International Conference on Robotics and Automation*, May 2019.

[Fallon *et al.*, 2014] Maurice F. Fallon, Matthew Antone, Nicholas Roy, and Seth Teller. Drift-free humanoid state estimation fusing kinematic, inertial and lidar sensing. In *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pages 112–119, 2014.

[Kwak *et al.*, 2009] Nojun Kwak, Olivier Stasse, Thomas Foissotte, and Kazuhito Yokoi. 3d grid and particle based slam for a humanoid robot. In *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pages 62–67, 2009.

[Mahony *et al.*, 2008] Robert Mahony, Tarek Hamel, and Jean-Michel Pflimlin. Nonlinear complementary filters on the special orthogonal group. *IEEE Transactions on Automatic Control*, 53(5):1203–1218, 2008.

[Mur-Artal and Tardós, 2017] Raul Mur-Artal and Juan D. Tardós. Orb-slam2: An open-source slam system for monocular, stereo and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.

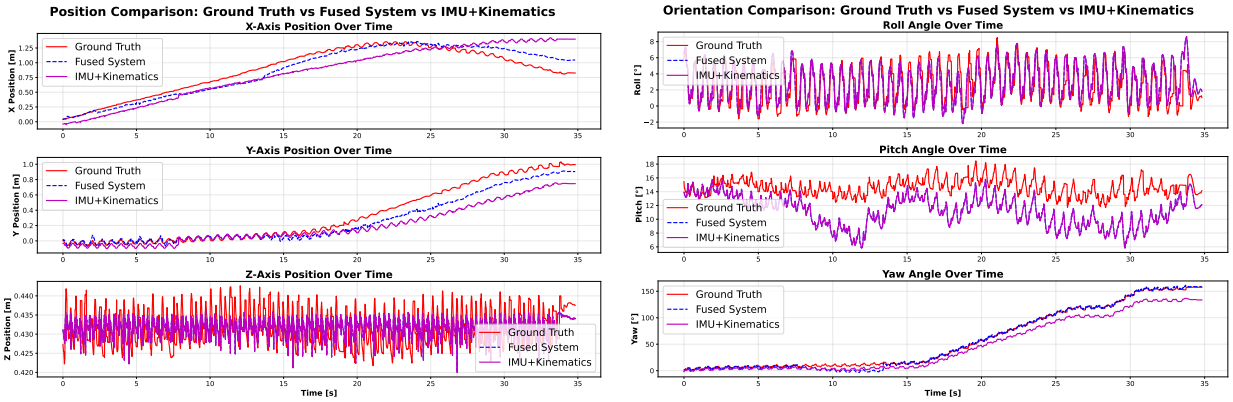
[O'Brien, 2024] Thomas O'Brien. Humanoid state estimation in RoboCup. In *Workshop on Humanoid Soccer Robots (WHSR 2024) at IEEE-RAS International Conference on Humanoid Robots (Humanoids 2024)*, Nancy, France, 2024. Extended abstract.

[Oriolo *et al.*, 2012] Giuseppe Oriolo, Antonello Paolillo, Luigi Rosa, and Marilena Vendittelli. Vision-based odometric localization for humanoids using a kinematic ekf. In *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pages 153–158, 2012.

[Oriolo *et al.*, 2016] Giuseppe Oriolo, Antonello Paolillo, Luigi Rosa, and Marilena Vendittelli. Humanoid odometric localization integrating kinematic, inertial and visual information. *Autonomous Robots*, 40(5):867–879, 2016.

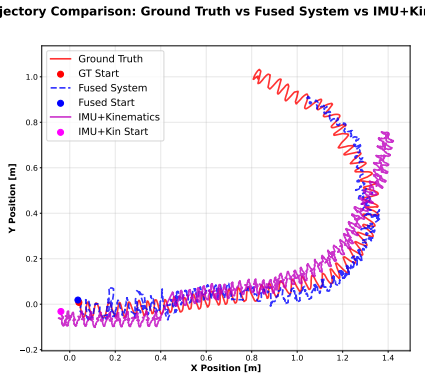
[Scona *et al.*, 2017] Raluca Scona, Simone Nobili, Yvan R. Petillot, and Maurice Fallon. Direct visual slam fusing proprioception for a humanoid robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1419–1426, 2017.

[Stasse *et al.*, 2006] Olivier Stasse, Andrew J. Davison, Rachid Sellaouti, and Kazuhito Yokoi. Real-time 3d



(a) Translation (x, y, z)

(b) Rotation (ϕ, θ, ψ)



(c) xy trajectory

Figure 7: Visual dropout experiment.

slam for a humanoid robot considering pattern generator information. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 348–355, 2006.

[Sumikura *et al.*, 2019] Shuji Sumikura, Mitsuki Shibuya, and Ken Sakurada. Openvslam: A versatile visual slam framework. In *ACM International Conference on Multimedia (MM)*, pages 2292–2295, 2019.

[Tsotsos *et al.*, 2012] Kostas Tsotsos, Alberto Pretto, and Stefano Soatto. Visual-inertial ego-motion estimation for humanoid platforms. In *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pages 704–711, 2012.