Hold Before You Step: The Information Requirement of In-Place Push Recovery on Force-Sensorless Humanoids

Jeonghyeon Kim[1], Jaesang Kim[2], and Shinsuk Park[1,*]

***Abstract*—In-place push recovery preserves a humanoid's task stance. Without force/torque sensors, the controller must decide whether to hold using a delayed and noisy estimate, but how much delay and error it can tolerate is unclear. We measure this information requirement on a simulated Unitree G1 by feeding a fixed hold-or-step controller the true push force, delayed and corrupted. In boundary tests, the initial loss of holdable force agreed with the pendulum rate of about 3.7 % per 10 ms; actuation delay drew on the same budget, and slow errors of 30 N were tolerated while 20 N errors on the push's time scale were not. The requirement guides an estimate that resolves the pelvis inertial measurement unit through the floating-base model, retains the joint accelerations and needs no upper-body torques; for three of four estimators, hold-entry latency predicted the normalised holdable force within 0.02 (registered tolerance 0.08). The resulting controller met its pre-registered recovery criterion against a stepping baseline with 49 % fewer steps and, on holdable pushes that both controllers recovered from, 96 % less base displacement. On the real G1, the estimate followed a load cell by 29 ms, against 57 ms for a momentum observer, and with steps disabled, the largest lateral push held per block had a median of 69 N with the hold layer, whereas without it no block held a push (12 blocks each). Disturbance estimates for balance should therefore be judged by the recovery they preserve.**

***Index Terms*—Humanoid robots, push recovery, disturbance estimation, legged locomotion, model predictive control.**

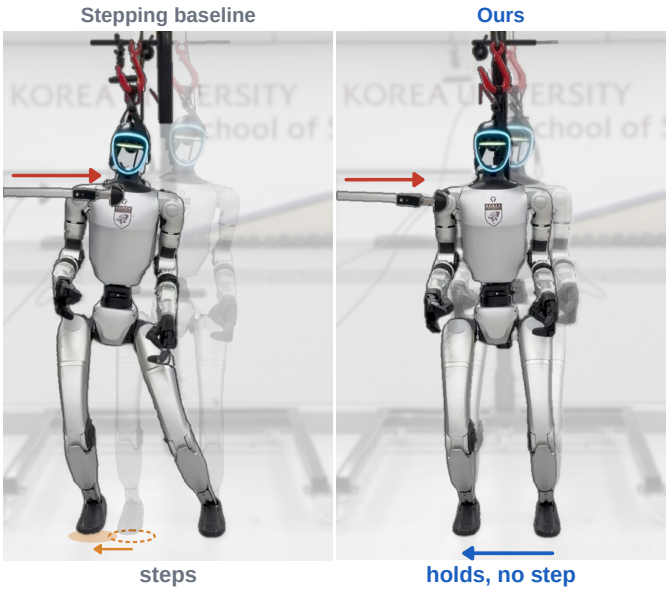

Fig. 1. A lateral push on the real G1: the stepping baseline steps, whereas our controller holds in place. Illustrative hand pushes in a safety gantry (not a registered evaluation; push forces not matched); poses 0 and 0.6 s (ours also 1.6 s) after contact, faint to solid; blue arrow: schematic ground reaction; quantitative results are in Sec. VI.

I. Introduction

HUMANOID robots must recover from disturbances while limiting disruption to their ongoing tasks. Recovery steps can interrupt manipulation performed from a fixed stance [1], and in human balance, effort favours in-place strategies when they suffice [2]. Recovery should therefore exploit the existing foot contacts and step only when they are insufficient (Fig. 1).

Existing controllers combine ankle and hip strategies with stepping, or switch between standing and stepping policies [1], [3]. Ankle action shifts the centre of pressure (CoP) and hip action regulates centroidal angular momentum within the current contacts, whereas stepping changes the support configuration [3], [4]. Capturability analysis characterises whether a given state can be brought to rest with a specified number of steps [5]. However, corrective CoP motion is bounded by the support region, so a delayed restoring response can allow motion caused by the disturbance to grow until a new contact becomes necessary.

Without dedicated force/torque (F/T) sensors, observers infer external loads from motion and actuation signals [6], and must also separate a transient push from sustained loads such as a carried object [7]. When these estimates determine when and how a controller responds, filtering can suppress noise at the cost of delaying the restoring action [8], while estimation errors can alter the decision to hold or step. The choice of estimator matters: on the same controller, a tuned momentum observer held a third as many pushes in place as our estimate (Sec. VI-D). Yet neither a state's capturability nor an estimate's accuracy alone establishes the delay and error that a specified controller can tolerate while retaining recovery without stepping.

We quantify this information requirement on a simulated Unitree G1 by supplying a fixed hold-or-step controller with delayed and corrupted true force. We then use the measured requirement to evaluate estimation and guide the controller's feedback paths. Our contributions are as follows (Fig. 2).

1) **The information requirement of in-place recovery.** We quantify how information delay and estimation error affect the maximum push force a fixed controller can reject without stepping. The initial loss of holdable force follows the rate predicted by an idealised linear inverted pendulum model. Information and actuation delays share a latency budget, while the tolerated error magnitude depends on the error's time scale.

[1]Jeonghyeon Kim and Shinsuk Park are with the Department of Mechanical Engineering, Korea University, Seoul, Republic of Korea (e-mail: ahdtld54@korea.ac.kr; drsspark@korea.ac.kr).

[2]Jaesang Kim is with the Department of Mechanical Engineering, University College London, London, U.K. (e-mail: jaesang8766@gmail.com).

*Corresponding author: Shinsuk Park.

2) **IMU-resolved external wrench estimation.** We reconstruct the resultant external force and moment, excluding foot–ground reactions, using a pelvis inertial measurement unit (IMU), joint encoders, and 12 leg torque estimates. Under stated model and contact assumptions, the floating-base reconstruction requires neither F/T sensors nor upper-body torque measurements. It retains joint acceleration terms and avoids differentiating reconstructed base velocity, reducing noise at matched latency relative to the tested momentum observers.
3) **Disturbance feedback for hold-or-step control.** We use the measured delay and error sensitivity to configure fast decision paths and filtered feedback for force-bound checks and CoP feedforward. In paired simulations, the controller meets the pre-registered recovery criterion against a stepping model predictive control (MPC) baseline with fewer steps and more pushes held in place. On holdable pushes recovered by both controllers, it also reduces base displacement and leg mechanical work. On the real G1, with steps disabled, the largest lateral push held per block had a median of 69 N with the hold layer, whereas without it no block held a push.

Recovery comparisons are conducted in simulation; hardware tests evaluate the estimators against a load cell and during the robot's own arm motion, and the hold layer against the same controller without it.

II. Related Work

Recovery strategies and the step decision. Capture-point model predictive control coordinates ankle, hip and stepping strategies [3], and a hierarchical scheme combines a learned standing policy and ankle modulation with a switch to stepping when instability is predicted [1]. State-based criteria for a necessary step include, besides capturability [5], the extrapolated centre of mass [9], the divergent component of motion (DCM) [10] and geometric sets of feasible centroidal moment pivots [11]. Standing controllers reject pushes without a step [12]; steps are triggered by planner feasibility [13], [14], predicted instability [15] or capture-point and velocity thresholds [16]. Learned policies recover from large pushes [17] without an explicit criterion for holding in place. These methods organise recovery from the measured state or an assumed disturbance; we ask what disturbance information the decision needs.

Timing and uncertainty in balance. Delayed state feedback stabilises the inverted pendulum only below a critical delay [18], and acceleration feedback counteracts reflex delay [19]. Bounded sensor and actuator uncertainties yield tracking-error bounds and sampling guidance for legged balance [20]. Push detectors trade lead time against false alarms [21], and an earlier recovery step improved push recovery [15]. Balance assistance must react faster than physiological responses to help [22], and information-theoretic bounds limit sensor-based control [23]. These concern the whole loop or human assistance; here only the push information is late or wrong, and, to our knowledge, no study varies its latency and error independently on a fixed controller and measures the in-place recovery that remains.

Disturbance estimation without F/T sensors. Feeding an external-force estimate back into a whole-body controller improved push recovery [24]. Generalised-momentum observers estimate external torques without joint accelerations [25], [26] and have been applied to humanoids with all joint torques [6]; enhanced observers add a lead term and a Kalman filter to trade phase lag against noise [8], and learned uncertainty torques, trained with foot F/T data, remove much of the model error [27]. Other estimators use measured contact forces with an IMU [28], [29], or resolve the IMU through a centroidal model without the joint accelerations [30], which carry the inertial force of moving arms. These estimators are evaluated by accuracy and response speed or by closed-loop recovery [31], not against the latency and error that an in-place hold tolerates. We keep the joint accelerations, show that the base and leg rows suffice without foot F/T sensors or upper-body torques (Proposition 2), and evaluate the estimators against that requirement.

III. The Information Requirement of an In-Place Hold

Definition 1 (Information-limited holdability)**.** Fix a controller C, an initial state, a contact set and actuator limits. Let C receive the push information delayed by L and corrupted by an error of standard deviation σ (white unless its spectrum is stated). The information-limited holdability $H_C(L, \sigma)$ is the fraction of a fixed set of pushes that C recovers without a step.

$H_C(0, 0)$ is an experimental reference for what the controller holds with ideal information, not a physical viability bound. We also measure the *holdable force*: the first downward crossing of a 50 % in-place recovery rate on a force grid, linearly interpolated between the adjacent forces (2.5 or 5 N spacing; three or four seeds per force). Boundaries outside the tested grid are censored and excluded from comparisons.

Proposition 1 (Holdable push versus hold-start latency)**.** Consider the linear inverted pendulum in the ground frame, $\dot{\xi} = \omega(\xi - p) + F/(m\omega)$, where the DCM ξ and the CoP p are measured from the stance centre and the support P is convex, closed and contains the stance centre ($P = [-b, b]$ along one direction). Let $\xi(0) = \xi_0$ and let a push $F(t)$ act on $[0, T]$, with the DCM displacement

$$\delta = \frac{1}{m\omega} \int_0^T e^{-\omega s} F(s)\, ds.$$

The CoP stays at $p = 0$ until the hold starts at $L \geq 0$ and is free in P afterwards. Within this model, a CoP schedule exists after which the state is 0-step capturable [5] iff

$$\xi_0 + \delta \in e^{-\omega L} P. \tag{1}$$

Hence, for $\xi_0 = 0$, the set of holdable pushes shrinks by the factor $e^{-\omega L}$ in every direction, for any push profile and duration. Along one direction, a constant push F is holdable iff

$$|F| \leq F_{\max}(L) = m\omega^2 b \, \frac{e^{\omega T}}{e^{\omega T} - 1} \, e^{-\omega L}. \tag{2}$$

Proof. Let $\tau = \max(T, L)$. Variation of constants gives

$$\xi(\tau) = e^{\omega \tau}(\xi_0 + \delta) - \int_L^\tau \omega e^{\omega(\tau - s)} p(s)\, ds.$$

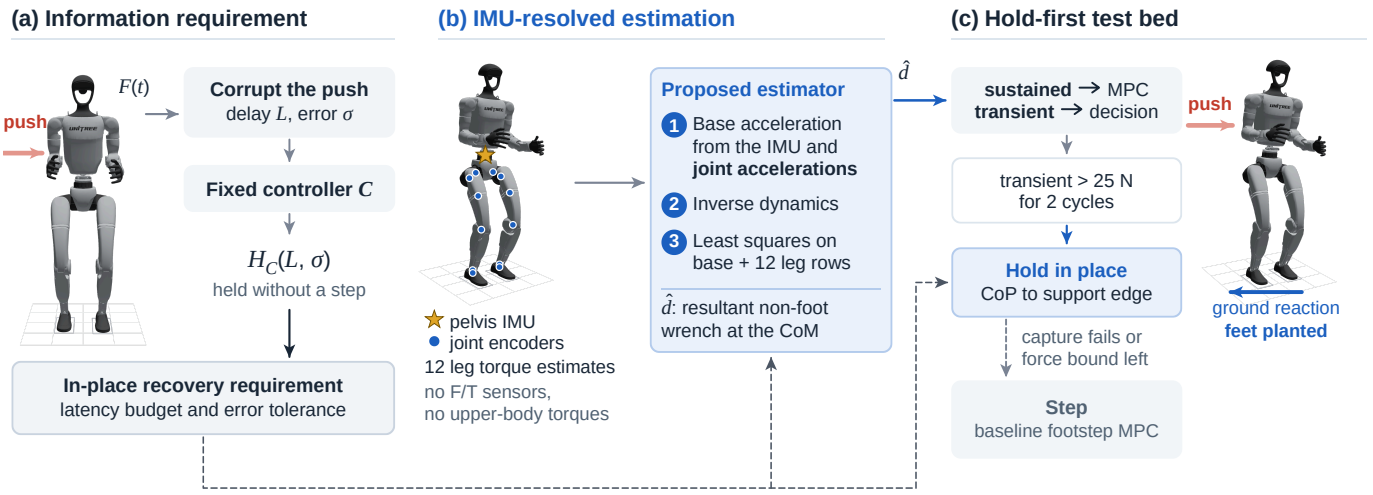

Fig. 2. Overview. (a) A fixed hold-or-step controller receives the true push force $F(t)$, delayed and corrupted. (b) The estimate $\hat{d}$ resolves the pelvis IMU through the floating-base model. (c) The hold-first test bed: the sustained part of $\hat{d}$ (0.5 s filter) loads the MPC; the transient part drives the hold decision. Dashed: the requirement guides the estimator and the signal paths.

The weights of the integral sum to $e^{\omega(\tau-L)}-1$, so, since P is convex and closed, the integral over all schedules in P takes exactly the values in $(e^{\omega(\tau-L)}-1)P$ (constant schedules attain each of them). Some schedule therefore ends in P iff $e^{\omega\tau}(\xi_0+\delta)\in P+(e^{\omega(\tau-L)}-1)P=e^{\omega(\tau-L)}P$, using $\lambda P+\mu P=(\lambda+\mu)P$ for convex P and $\lambda,\mu\geq 0$; this is (1). If $\xi(\tau)\in P$, the CoP $p=\xi(\tau)$ keeps $\dot{\xi}=0$, so the robot stops without a step. For a constant push, $\delta=F(1-e^{-\omega T})/(m\omega^2)$, which gives (2). □

The proposition extends 0-step capturability [5], [11] to late push information. On the G1 ($\omega=3.79\,\mathrm{s}^{-1}$), each 10 ms of latency costs about 3.7 % of the holdable force.

The model ignores angular momentum and foot rotation, and stance feedback before the hold shifts ξ_0, so $e^{-\omega L}$ is exact only without pre-hold stance feedback; we therefore test the ratio, not the absolute boundary, on constant 0.2 s pushes.

Two consequences follow within the model. First, a command delay D_a that postpones the CoP schedule replaces L by $L+D_a$: information and actuation delay share one budget. Second, a hand load or own arm motion that reserves part of the support restricts the CoP to a convex, closed $P_r\subseteq P$ containing the stance centre, for which the proof holds unchanged; along one direction, with $P_r=[-(b-b_r),b-b_r]$ and $\xi_0=0$,

$$|\delta|\leq e^{-\omega L}\,(b-b_r), \tag{3}$$

so the delay factor multiplies whatever margin remains.

IV. An Estimate That Meets the Requirement

The requirement asks for an estimate that is fast at low noise. A momentum observer differentiates, in effect, a base velocity reconstructed from leg kinematics; we instead measure the base acceleration with the pelvis IMU and resolve it through the model. The floating-base dynamics are

$$M(q)\ddot{q}+h(q,\dot{q})=S^{\top}\tau+\textstyle\sum_i J_i^{\top}f_i+\sum_k J_k^{\top}w_k, \tag{4}$$

with foot wrenches f_i and non-foot external wrenches w_k. We partition $q=(q_b,q_L,q_U)$ into the floating base (the pelvis), the 12 leg joints and the 17 upper-body joints. Let d be the resultant wrench of all w_k, expressed at the centre of mass (CoM).

Estimate. Each control cycle has three steps. First, the pelvis IMU's specific force a_s and differenced angular rate are affine in the base acceleration for given joint accelerations, $[\dot{\omega}_s;a_s]=J_s\ddot{q}+\dot{J}_s\dot{q}-[0;R_s^{\top}g]$. We solve these six rows for the base acceleration, with the joint accelerations from control-cycle differences of the estimated joint velocities. Second, we evaluate the inverse dynamics. Third, we solve the 6 base and 12 leg rows of (4) by unweighted, unregularised least squares for d (at the CoM) and the 6D wrench of each foot in contact (at the ankle), using the leg torque estimates τ_L with the model's leg friction subtracted. A first-order filter of bandwidth $K=100\,\mathrm{s}^{-1}$, $y_k=y_{k-1}+\min(K\Delta t,1)\,(x_k-y_{k-1})$, smooths the result. The estimate uses no upper-body torque, but the positions and differenced velocities of all joints. It relies on current-based leg torques, whose errors under dynamic loading remain to be characterised.

Proposition 2 (Lower-body sufficiency)**.** Assume (A1) the non-foot wrenches act on the pelvis or the upper body, not on the legs; (A2) each leg in contact has six joints and a non-singular foot Jacobian $J_{i,L}$, and each foot wrench is a full 6D wrench; (A3) the generalised position, velocity and acceleration of all joints and the base are available, and the model is exact. Then the base and leg rows of (4) determine the foot wrenches and the resultant d uniquely, and τ_U does not enter. Only the resultant, not the location or distribution of the non-foot wrenches, is identifiable.

Proof. In a kinematic tree the row of joint j in $J_k^{\top}$ is non-zero only if body k is distal to j. Feet are distal only to their own leg joints. By (A1), the leg rows contain no non-foot wrench and read $M_L\ddot{q}+h_L=\tau_L+J_{i,L}^{\top}f_i$ per leg, and by (A2) $J_{i,L}$ is invertible, which fixes f_i. The base rows carry

no actuation and are the whole-body Newton–Euler equations $\dot{h}_G = \sum_i \mathrm{Ad}_i^\top f_i + d + mg$ about the CoM, with the centroidal momentum h_G, the transforms $\mathrm{Ad}_i^\top$ of the foot wrenches to the CoM and the gravity wrench mg; they fix d. □

Upper-body model and state errors still enter through $M\ddot{q}$ and h, and (A2) fails near a leg singularity (condition number below 70 in our runs).

Proposition 3 (Noise scaling of the two structures, first-order approximation)**.** Model the base-velocity and joint-velocity errors as zero-mean noise with standard deviations σ_{v_b} and $\sigma_{\dot{q}}$ whose power lies mostly at frequencies well above K, and the accelerometer error as white noise of density σ_a. Pass each raw estimate through a first-order filter of bandwidth K and neglect sampling ($K\Delta t \ll 1$). For a fixed disturbance amplitude and detection threshold, the detection latency of a step disturbance then scales with $1/K$. (i) The noise of a floating-base momentum observer has standard deviation $\approx K\, m\, \sigma_{v_b}$, with the total mass m. (ii) The noise of the IMU-resolved estimate has a term $\propto \sqrt{K}\, m\, \sigma_a$ and, for each joint, a term $\propto K\, m_{\mathrm{dist}}\, \sigma_{\dot{q}}$, with the mass m_{dist} distal to that joint.

Proof. With the generalised momentum $\pi = M\dot{q}$ and $\dot{\pi} = \tau + \beta + \tau_{\mathrm{ext}}$, the observer residual $r = K(s\pi - \beta - \tau)/(s+K)$ passes a momentum error as $Ks\,\delta\pi/(s+K)$, a high-pass of gain K that transmits nearly all the variance of a velocity error with power well above K, giving $\approx K\, m\, \sigma_{v_b}$. In the IMU-resolved estimate, white accelerometer noise passes through $K/(s+K)$ with standard deviation $\sigma_a\sqrt{K/2}$, and differenced joint velocities enter through the momentum-Jacobian columns, which scale with the distal mass, and the same high-pass, giving $\approx K\, m_{\mathrm{dist}}\, \sigma_{\dot{q}}$. □

Where the base-velocity error dominates, these relations predict a noise advantage at equal latency that grows as latency shrinks, the range Proposition 1 singles out; the joint-velocity term or sampling ($K\Delta t = 0.5$ at 200 Hz) can dominate instead, so Section VI-B measures noise at matched latency.

V. Test Bed: A Hold-First Controller

The test bed isolates the hold-or-step decision: an estimate-driven hold layer is added to a stepping baseline, which consists of a centroidal model predictive controller (MPC), a footstep MPC that chooses the step location and swing duration, and a whole-body inverse-dynamics quadratic program (QP). The estimate is split into a sustained part (0.5 s filter), which the baseline treats as a load, and a transient part d_{fast}. When the horizontal force of d_{fast} exceeds 25 N for two control cycles, the controller enters the hold mode, an ankle-priority mode, before the planner reacts. In this mode, a linear program over the QP's variables drives the CoP to the support edge to cancel the transient; it limits the centroidal angular-momentum rate and maintains at least 10 % of the weight on each foot. The hold is thus designed to realise the CoP schedule of Proposition 1 after the hold-entry latency L, which includes the two confirmation cycles and is measured in Sec. VI-B. Steps are blocked during a hold. The hold ends and a step is triggered when a 0-step capture prediction fails (linear inverted pendulum over 0.3 s, CoP at the support edge toward the DCM) or when the transient leaves the force bound, an ellipse of 60 N fore-aft and 75 N lateral (45 N fore-aft while the arms are commanded to move). All constants were fixed on development seeds before the evaluations.

Two configurations. This *evaluation configuration* measures the requirement and serves the offline comparison; the *final configuration* (Sec. VI-C), used for the closed-loop estimator boundaries and all controller comparisons, changes the information paths and, during arm motion, the hold reference.

VI. Experiments

Setup. A Unitree G1 with 29 degrees of freedom is simulated in MuJoCo [32] at 200 Hz with encoder, torque-estimate and IMU noise, a 10 ms observation delay and the IMU at the pelvis, as on the hardware; pushes of 0.2 s act on the torso. A run *fails* if a body other than the feet touches the ground, the pelvis drops below 0.5 m or a swing foot does not touch down (a tilt stop at 1.0 rad terminates divergent runs); it is *recovered* if it does not fail and ends standing (last 0.5 s: pelvis above 0.6 m, mean speed below 0.1 m/s), and *held in place* if recovered without a step.

Protocol. Every comparison was pre-registered with a recorded hash before its runs on new seeds (offline comparisons: before the analysis), and no decision threshold was changed afterwards, except for the outcome-definition amendment below; analyses marked post hoc were not registered. A registered amendment replaced the original 0.35 rad tilt stop, which also ended standing runs during arm lifts, and re-scored all 4372 runs (40 to recovered, 22 to failed, two decisions changed; counts were stable for tilt stops of 0.6–1.2 rad). One-sided tests in the registered direction are exact McNemar (paired recovery, same condition and seed), Wilcoxon signed-rank (paired estimator metrics, steps) and Mann–Whitney (hardware blocks). The p-values are nominal and unadjusted except for the Holm-corrected E2 follow-up; runs sharing a seed share sensor noise and torque-estimate biases (four or six seeds per cell). Registrations, analysis code and run summaries: https://juunghyeon.github.io/hold-before-you-step/registrations/.

Timing. The true force bypasses the 10 ms observation delay ($L = 0$). *Hold-entry latency* runs from push onset to the first hold cycle; offline *detection latency* ends when the filtered estimate ($K = 100\,\mathrm{s}^{-1}$) exceeds 25 N on two consecutive samples.

A. The information requirement

Near the boundary, holding is lost within the first tens of milliseconds. We fixed the controller and replaced the estimate in its whole hold decision (entry, ankle target, capture prediction and force bound) by the true push force, delayed by L and corrupted by white noise σ per axis, on pushes near the boundary (± 70 N lateral, ± 50 N fore-aft and diagonal; Fig. 3a). With ideal information, 10 of 12 pushes were held; a delay of 10 ms left 10 and one of 20 ms left 3. Neither registered monotonicity prediction held over the full grid: 20 N of noise suppressed holding even without delay, whereas moderate noise

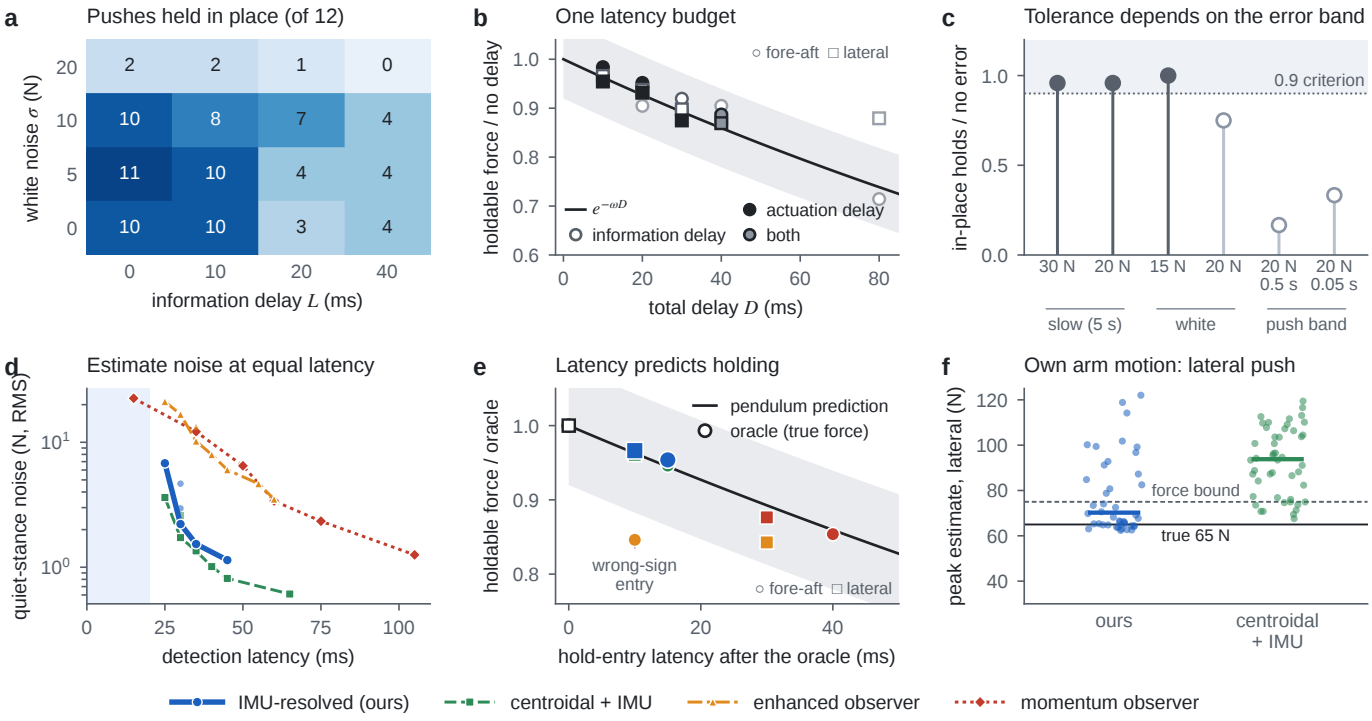

Fig. 3. The information requirement of the hold (top) and the estimators against it (bottom). (a) Pushes held without a step when the true push force reaches the hold decision delayed by L and with white noise σ (12 near-boundary pushes per cell; 95 % Wilson intervals at $\sigma = 0$: 55–95 % at 0 ms, 9–53 % at 20 ms). (b) Holdable force relative to no delay versus the combined actuation and information delay D (608 runs) and, hollow markers, versus information delay with the capture prediction as the only step trigger (boundary search); line: $e^{-\omega D}$, band ± 0.08. (c) Holds under injected errors of different correlation times, relative to no error (240 runs); filled: within the registered tolerance of 0.9 (shaded). (d) Quiet-stance noise versus detection latency over each estimator family's tuning range (55 logs; lines: lowest noise at each latency, pale points: all settings). (e) Closed-loop holdable force against the prediction from hold-entry latency (380 runs); circles fore-aft, squares lateral. (f) Own arm motion, post hoc: peak estimate of each 65 N lateral boundary push (dots; bar: median), ours against the centroidal estimate; dashed: lateral force bound. Shading: ± 0.08 tolerance in (b) and (e), first 20 ms in (d).

at 20 ms raised the holds (3 to 4 and 7) by triggering entry earlier. Our estimate held 7 of 12.

The loss follows the pendulum rate. With the capture prediction as the only step trigger, a boundary search (Fig. 3b, light markers) found the normalised holdable force within ± 0.08 of $e^{-\omega L}$ (about one 5 N step of the force grid) at 20 and 40 ms in both directions and at 80 ms fore-aft. Laterally, the boundary stopped falling after 40 ms, when the stance feedback on the wider support became dominant. The four tested delays support only the initial rate.

Actuation delay draws on the same budget. Delaying the joint commands to the drives by 10–30 ms cost as much holdable force as an equal information delay (largest difference 0.024 of the no-delay boundary; tolerance 0.05), and every combination followed the rate of the total delay within ± 0.03 (608 runs; tolerance ± 0.08; Fig. 3b).

The rate holds under hand loads and own arm motion. With a 2 kg hand load, arm motion or both (1200 runs), the normalised holdable force matched $e^{-\omega L}$ in all 24 checks (largest deviation 0.05), supporting the ratio in (3).

Which errors matter depends on their time scale. We added Ornstein–Uhlenbeck errors of equal stationary standard deviation but different correlation times to the true force from the run start (240 runs; Fig. 3c). Slow errors of 20 and 30 N (correlation time 5 s) left 23 of 24 near-boundary pushes held in place (24 without error), because the hold entry acts on the estimate minus its sustained part; an error of 20 N on the time scale of a push (0.5 and 0.05 s) left 4 and 8. White noise was tolerated at 15 N (24 of 24; registered after the 20 N result) but not at 20 N (18), where it kept the hold mode active in quiet stance. This brackets this controller's tolerance for white noise only, excluding motion-induced and endogenous errors.

B. Estimators against the requirement

We compared ours with three alternatives: a momentum observer using all 29 joint torques [25], [26] (gain $30\,\mathrm{s}^{-1}$ with a 0.1 s output filter, tuned in closed loop), our re-implementation of the enhanced momentum observer with a Kalman filter [8] (measurement noise matched to ours), and a centroidal estimate that is also IMU-resolved but omits joint accelerations [30].

Offline, ours is faster at lower noise, consistent with Proposition 3. On the same pelvis-IMU logs of 55 pushes (30–70 N along x, y, xy), ours detected pushes after 32 ms on average against 70 ms for both momentum observers, with the lowest push-phase error at every magnitude and direction (root-mean-square, RMS, 19.6 against 21.3–28.2 N; paired Wilcoxon $p < 0.002$). Across the tuning sweeps (Fig. 3d), the observer-to-ours noise ratio at matched latency rose from 1.1 at 105 ms to 7.9 at 35 ms. The ranking held at twice the sensor noise; at the highest level, ours and the centroidal estimate did not differ significantly (21.2 against 21.8 N, $p = 0.2$).

In closed loop, latency predicted holding. Proposition 1 predicts a normalised holdable force of approximately $e^{-\omega \Delta L}$ from the hold-entry latency ΔL relative to the true-force oracle, provided the estimation error remains within tolerance. In the final configuration (380 runs; Fig. 3e), ours entered after 10–15 ms and kept 0.95–0.97 of the true-force boundary, the tuned momentum observer entered after 30–40 ms and kept 0.85–0.88, and both matched the prediction within 0.017 (tolerance 0.08), as did the centroidal estimate (0.95–0.96). The prediction, ω and the tolerance were fixed in the registrations; nothing was fitted to the estimator boundaries. Fore-aft, the enhanced observer reached 0.85 instead of the predicted 0.96. A wrong-sign transient of 21–32 N, about 10 ms after each onset, started holds in the wrong direction. Post hoc bootstrap 95 % intervals of the deviation (runs resampled within force levels, boundaries and latencies recomputed; also with seeds as clusters) lay within ± 0.05 for the other three estimators and at -0.15 to -0.02 for the enhanced observer, excluding the force-grid spacing and the boundary interpolation.

Joint accelerations matter when the robot moves its arms. The two IMU-resolved estimates differ only in the joint-acceleration terms, which carry the inertial force of the arms. At the in-place boundary during four motion-capture arm motions (± 40 N fore-aft, ± 65 N lateral; 96 pairs), ours held 63 of the 96 pushes in place against 32 ($p < 10^{-7}$), recovered 85 against 67 runs and took 62 against 153 steps. Post hoc inspection showed that the centroidal method overestimated the 65 N lateral pushes (median peak 94 against 70 N; Fig. 3f), crossing the force bound and releasing the hold. With 50 N pushes, ours held 30 of 48 in place against 23 ($p = 0.008$), under the amended outcome definition.

C. Requirement-guided design

Motivated by the measured sensitivity to delay, the paths are treated differently. The *decision* (hold entry, capture prediction) uses the unfiltered estimate and a one-cycle entry instead of the filter and two-cycle confirmation; the *force bound* uses a five-cycle median that rejects brief IMU spikes (up to 96 N during a 60 N push) that otherwise triggered unnecessary steps; the *actuation* (CoP feedforward) keeps the filter, because it closes a loop through the accelerations the estimate uses and oscillated at the Nyquist frequency without it. Without filtering on any path, the controller held more single pushes (20 to 28, $p = 0.011$) but recovered fewer successive pushes and stepped more during arm motion. During commanded arm motion, the hold also keeps the MPC's planned angular-momentum rate and vertical motion as its reference. Against the evaluation configuration (104 runs each), the final configuration raised single-push holds from 24 to 31 ($p = 0.008$) and cut the steps by 37 %; the comparison evaluates these changes together.

D. Final comparison

The recovery criterion was met with half the steps (Table I). The registered criteria allowed at most 4 fewer recovered runs of the 80 and at most 80 % of the steps (pass criteria, not a non-inferiority test). Against the stepping baseline, ours recovered 65 against 62 runs, held 30 pushes in place against none and took 49 % fewer steps. Because failed runs end at different times, we also compared steps on the runs both recovered (post hoc): 34 against 89 ($p < 10^{-7}$). The gain lies in single pushes (50 against 45 of 56); on the 24 successive-push runs, no controller held in place, and ours recovered 15 against 17.

TABLE I

FINAL COMPARISON (80 PAIRED RUNS: 50–80 N SINGLE AND SUCCESSIVE PUSHES). ALL ROWS SHARE THE PLANT, STANCE CONTROLLER, FOOTSTEP MPC AND PUSHES. BASELINE: FEASIBILITY TRIGGER WITH STEP-TIMING MPC. LITERATURE RULES: OUR IMPLEMENTATIONS ON OUR HOLD LAYER, NOT THE PUBLISHED CONTROLLERS; LEARNED POLICIES [17] WERE NOT COMPARED. ZERO-STEP: RECOVERED WITHOUT A STEP.

Controller	Recovered	Zero-step	Steps
Ours	**65**	**30**	**117**
Stepping baseline	62	0	228
Predicted instability [15]	65	13	165
Capture point + velocity [16]	51	32	120
0-step capture gate [5], [33]	48	32	151
Ours, momentum observer	63	10	181

Both the step decision and its information matter. With the step-triggering rules from the literature on the same hold layer, ours recovered more runs than the two capture-point rules ($p \leq 0.008$) and, against the predicted-instability rule, met the recovery criterion with 29 % fewer steps and more than twice the pushes held in place. With the tuned momentum observer in place of our estimate, the same controller met the recovery criterion but took 181 instead of 117 steps and held 10 instead of 30 pushes in place.

The advantage persists during the robot's own arm motion and under hand loads. Over 48 such runs, ours recovered 45 against 39, taking 30 instead of 72 steps. With a 2 kg hand load during arm motion (288 runs), ours held 30 of the 48 runs with a 50 N push in place, against one for the baseline ($p < 10^{-8}$).

E. Cost of recovery

Holding in place is also cheaper. A registered comparison (348 runs) measured costs over 2.9–6.0 s (push at 3 s) on runs both recovered, as median paired reductions (thresholds 50, 15 and 25 %). On holdable pushes (30–50 N; 60 runs), ours held 59 in place against 12 for the baseline and reduced the net horizontal base displacement by 96 % (medians 1 against 62 mm), the peak foot force by 44 % and the leg work $\sum_j \int |\tau_j \dot{q}_j| \, dt$ over the 12 leg joints by 69 %. With hand loads and objects picked and placed between pushes (48 runs), ours held all 48 in place, whereas the baseline recovered 46 with 82 steps. Because the baseline stepped even at ± 30 N fore-aft, these savings reflect that stepping and concern motion and loading, not task performance.

F. Hardware validation

The G1 stood in a gantry, with the IMU at the pelvis as in simulation. Pushes were applied by hand through a load cell (7.5 kHz) logged on the robot's clock. For E1 and E2, the learned policy [17] kept the robot standing and moved its arms, while our controller computed its commands without sending them. The estimators were replayed causally using the 1 kHz

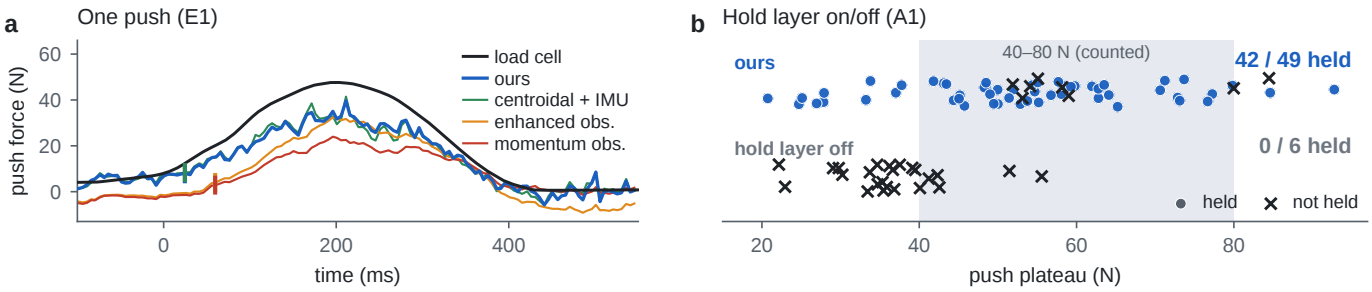

Fig. 4. Hardware, registered tests. (a) E1: one push from behind, load-cell force (10 ms mean) and the four estimates along the push direction, replayed causally at 200 Hz; ticks: each signal's 50 % crossing of its own plateau. Ours and the centroidal estimate nearly coincide on a push; they differ in the joint-acceleration terms, which mattered during arm motion in simulation (Fig. 3f) and in the delay-tuned hardware test (E2). (b) A1, steps disabled: every counted lateral push by its load-cell plateau, held (dots) or not (crosses), for ours and the same controller with the hold layer off; counts: held of the 40–80 N pushes (shaded).

TABLE II
HARDWARE ESTIMATORS AT THEIR OPERATING SETTINGS (E1, MEDIANS). DELAY: BEHIND THE LOAD CELL AT 50 % OF THE PLATEAU; ERROR, NOISE: PUSH-PHASE AND QUIET-STANCE RMS.

Estimator	Delay (ms)	Error (N)	Noise (N)
IMU-resolved (ours)	**29**	8.4	2.7
Centroidal + IMU	34	8.0	2.3
Enhanced observer	63	8.6	2.4
Momentum observer	57	11.0	2.2

joint and IMU logs at the 200 Hz control rate with the code and settings of the simulation comparisons. Metrics, exclusion rules (load cell and foot contact only) and tests were registered after a pilot session; the 71 pushes of 16–45 N came from two sessions, in four directions.

Detection latency and error (E1). As hand pushes vary, latency was measured between the load cell's and each estimate's crossing of 50 % of its own plateau (gain-normalised). Our estimate responded earlier than the momentum observer (paired difference 24 ms, ours earlier in 60 of 69 pushes, $p = 7 \times 10^{-11}$; 2 pushes without a crossing excluded; Table II, Fig. 4a), with a lower push-phase error but a slightly higher quiet-stance noise; at these unmatched operating settings, the matched-latency noise comparison is left to E2.

Own arm motion (E2). During four scripted arm motions without a push, the peak spurious estimate at the registered bandwidth was 43 N for ours and 40 N for the centroidal estimate (medians); ours was lower in 25 of 42 repetitions ($p = 0.087$), so the registered test was not met. Post hoc, the pelvis IMU showed five to ten times its quiet vibration above 15 Hz, which a faster estimate transmits, so we registered a second test on new recordings (two sessions, 40 repetitions), filtering both IMU-resolved estimates at 25 s^{-1}, the rate chosen to approximate the observer's E1 delay. The peak was 15.7 N for ours against 18.9 N for the centroidal estimate (lower in 34 of 40) and 18.4 N for the momentum observer (lower in 23 of 40, not during the lift: 21.7 against 24.4 N). Both predictions were met after Holm correction (unadjusted $p = 6 \times 10^{-6}$ and $p = 0.048$). On pushes recorded in the same sessions, ours still responded earlier than the momentum observer (55 against 66 ms). At this setting, ours read the least spurious force, consistent with Proposition 3; at its operating bandwidth it trades this margin for an earlier response.

Holding in place. With our controller in command and steps disabled, 21 of 22 lateral hand pushes of 26–59 N (0.25–1.2 s) were held without a step, under a registered outcome rule (no foot lift, no pelvis tilt change above 8°, no straightened knee and no guard stop within 3 s of onset). The hold mode started a median of 69 ms after the load-cell force crossed 25 N (interquartile range 59–144 ms). These pushes are the no-delay arm of a registered latency test that was stopped (Sec. VII). The rounded sole may reduce the usable CoP margin, an effect not quantified here.

Hold-layer ablation (A1). A registered test alternated blocks of ours and of the same controller with only the hold entry disabled, both with steps disabled. Each condition had 12 blocks in two sessions, each an ascending series of shoulder-height lateral pushes (aimed 30–80 N) ended by a failure; held as above, with a tilt change below 12°. The primary measure was the block score, the largest held plateau (capped at 80 N; 0 for a block without a held push). Its median was 69 N for ours and 0 N for the ablation, which held no counted push in any block (one-sided Mann–Whitney, $p = 5 \times 10^{-6}$, blocks treated as independent; each session alone: 62 and 75 N against 0 N; unchanged with the 3 s window from release instead of onset). Because blocks ended at a failure, the ablation met fewer high forces, so the per-push counts are descriptive: between 40 and 80 N, ours held 42 of 49 pushes and the ablation 0 of 6 (Fig. 4b). The ablation unloaded a foot in 23 of 24 pushes (22–56 N), consistent with the hold preserving contact; ours, keeping 10 % on each foot, failed at 52–84 N. A registered test with sustained (2 s) pushes was inconclusive, as push duration depended on the outcome; descriptively, the ablation unloaded a foot in all 23 pushes and ours held 19 of 38.

VII. CONCLUSION

This work quantified, for a fixed controller, the disturbance information needed for recovery without stepping, making recovery capability an estimator design criterion. Controlled simulations revealed a shared latency budget for information and actuation, while errors varying during a push were more disruptive than slow errors. An IMU-resolved wrench estimate gave timely feedback without F/T sensors, with lower noise than momentum observers at matched latency. This requirement predicted, from hold-entry latency alone, the normalised

holdable force of three of four estimators without fitting; the fourth showed that the sign of early errors also matters. Guided by this requirement, with fast decision paths and filtered feedback, the controller met its pre-registered recovery criterion with 49 % fewer steps than the stepping baseline. On the real G1, the estimate followed a load cell by 29 ms against 57 ms for the momentum observer. With stepping disabled, a registered ablation confirmed the hold layer's benefit for holding lateral pushes in place.

A key limitation is that hand-push variability prevented us from resolving the predicted delay effect on hardware, and recovery with stepping was not evaluated there. Future work should test this effect with a repeatable pusher and combine the hold layer and its information requirement with learned stepping policies such as [17]. More broadly, stating the information that an in-place hold needs lets estimators and feedback paths be compared by the recovery they preserve, not by accuracy alone.

REFERENCES

[1] C. Zhang, J. Bai, Z. Chen, J. Gao, and H. Qiao, "Hierarchical control for robust standing stability and fall recovery of task-performing humanoid robots," *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 20 920–20 933, 2025.

[2] M. Afschrift, I. Jonkers, J. De Schutter, and F. De Groote, "Mechanical effort predicts the selection of ankle over hip strategies in nonstepping postural responses," *Journal of Neurophysiology*, vol. 116, no. 4, pp. 1937–1945, 2016.

[3] M.-J. Kim, D. Lim, G. Park, K. Lee, and J. Park, "A model predictive capture point control framework for robust humanoid balancing via ankle, hip, and stepping strategies," *IEEE Transactions on Robotics*, vol. 41, pp. 3297–3316, 2025.

[4] F. B. Horak and L. M. Nashner, "Central programming of postural movements: adaptation to altered support-surface configurations," *Journal of Neurophysiology*, vol. 55, no. 6, pp. 1369–1381, 1986.

[5] T. Koolen, T. de Boer, J. Rebula, A. Goswami, and J. Pratt, "Capturability-based analysis and control of legged locomotion, Part 1: Theory and application to three simple gait models," *The International Journal of Robotics Research*, vol. 31, no. 9, pp. 1094–1113, 2012.

[6] F. Flacco, A. Paolillo, and A. Kheddar, "Residual-based contacts estimation for humanoid robots," in *IEEE-RAS International Conference on Humanoid Robots*, 2016, pp. 409–415.

[7] H. Song, W. Z. Peng, and J. H. Kim, "Balancing of humanoid with object mass: trade-off analyses and lifting control," arXiv:2607.29625, 2026.

[8] J. Sun, Z. Wang, Y. Gao, R. Huang, Y. Li, and L. Zhou, "Proprioceptive estimation of external wrenches for legged robots using enhanced momentum-based Kalman filter," *IEEE Transactions on Instrumentation and Measurement*, vol. 75, p. 7506812, 2026.

[9] A. L. Hof, M. G. J. Gazendam, and W. E. Sinke, "The condition for dynamic stability," *Journal of Biomechanics*, vol. 38, no. 1, pp. 1–8, 2005.

[10] J. Englsberger, C. Ott, and A. Albu-Schäffer, "Three-dimensional bipedal walking control based on divergent component of motion," *IEEE Transactions on Robotics*, vol. 31, no. 2, pp. 355–368, 2015.

[11] R. Schuller, G. Mesesan, J. Englsberger, C. Ott, J. Lee, and A. Albu-Schäffer, "Grounding the three-dimensional divergent component of motion: Geometric analysis of contact and dynamic stability and its application to humanoid push recovery," *The International Journal of Robotics Research*, 2026, early access.

[12] J. A. Castano, C. Zhou, Z. Li, and N. Tsagarakis, "Robust model predictive control for humanoids standing balancing," in *International Conference on Advanced Robotics and Mechatronics (ICARM)*, 2016, pp. 147–152.

[13] M. Khadiv, A. Herzog, S. A. A. Moosavian, and L. Righetti, "Walking control based on step timing adaptation," *IEEE Transactions on Robotics*, vol. 36, no. 3, pp. 629–643, 2020.

[14] F. M. Smaldone, N. Scianca, L. Lanari, and G. Oriolo, "Feasibility-driven step timing adaptation for robust MPC-based gait generation in humanoids," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1582–1589, 2021.

[15] J. Li, Z. Le, J. Ma, and Q. Nguyen, "Adapting gait frequency for posture-regulating humanoid push-recovery via hierarchical model predictive control," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2025, pp. 12 330–12 336.

[16] X. Meng, Z. Yu, T. Han, X. Liu, Q. Li, Q. Huang, X. Chen, and F. Meng, "Adaptive disturbance rejection balance control for humanoid robots via variable-inertia centroidal MPC," *Journal of Bionic Engineering*, vol. 22, pp. 2885–2899, 2025.

[17] Y. Zhang, Y. Yuan, P. Gurunath, T. He, S. Omidshafiei, A.-a. Aghamohammadi, M. Vazquez-Chanlatte, L. Pedersen, and G. Shi, "FALCON: Learning force-adaptive humanoid loco-manipulation," arXiv:2505.06776, 2025.

[18] J. Milton, J. L. Cabrera, T. Ohira, S. Tajima, Y. Tonosaki, C. W. Eurich, and S. A. Campbell, "The time-delayed inverted pendulum: Implications for human balance control," *Chaos*, vol. 19, no. 2, p. 026110, 2009.

[19] T. Insperger, J. Milton, and G. Stépán, "Acceleration feedback improves balancing against reflex delay," *Journal of the Royal Society Interface*, vol. 10, no. 79, p. 20120763, 2013.

[20] N. A. Villa, J. Englsberger, and P.-B. Wieber, "Sensitivity of legged balance control to uncertainties and sampling period," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3665–3670, 2019.

[21] R. Subburaman, D. Kanoulas, N. Tsagarakis, and J. Lee, "A survey on control of humanoid fall over," *Robotics and Autonomous Systems*, vol. 166, p. 104443, 2023.

[22] O. N. Beck, M. K. Shepherd, R. Rastogi, G. Martino, L. H. Ting, and G. S. Sawicki, "Exoskeletons need to react faster than physiological responses to improve standing balance," *Science Robotics*, vol. 8, no. 75, p. eadf1080, 2023.

[23] A. Majumdar, Z. Mei, and V. Pacelli, "Fundamental limits for sensor-based robot control," *The International Journal of Robotics Research*, vol. 42, 2023.

[24] J. Li, Z. Yuan, S. Dong, J. Zhang, and X. Sang, "External force observer aided push recovery for torque-controlled biped robots," *Autonomous Robots*, vol. 46, no. 5, pp. 553–568, 2022.

[25] A. De Luca and R. Mattone, "Sensorless robot collision detection and hybrid force/motion control," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2005, pp. 999–1004.

[26] S. Haddadin, A. De Luca, and A. Albu-Schäffer, "Robot collisions: A survey on detection, isolation, and identification," *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1292–1312, 2017.

[27] D. Lim, M.-J. Kim, J. Cha, and J. Park, "MOB-Net: Limb-modularized uncertainty torque learning of humanoids for sensorless external torque estimation," *The International Journal of Robotics Research*, vol. 44, no. 1, pp. 96–128, 2025.

[28] M. Benallegue, P. Gergondet, H. Audren, A. Mifsud, M. Morisawa, F. Lamiraux, A. Kheddar, and F. Kanehiro, "Model-based external force/moment estimation for humanoid robots with no torque measurement," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 3122–3129.

[29] V. Morlando and F. Ruggiero, "Disturbance rejection for legged robots through a hybrid observer," in *30th Mediterranean Conference on Control and Automation (MED)*, 2022.

[30] C. Fedsi, M. Mallem, and M. Guiatni, "Force/torque sensor-free framework for external force estimation in humanoid robots," *Advanced Robotics*, 2026.

[31] V. Morlando, A. Teimoorzadeh, and F. Ruggiero, "Whole-body control with disturbance rejection through a momentum-based observer for quadruped robots," *Mechanism and Machine Theory*, vol. 164, p. 104412, 2021.

[32] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A physics engine for model-based control," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 5026–5033.

[33] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: A step toward humanoid push recovery," in *IEEE-RAS International Conference on Humanoid Robots*, 2006, pp. 200–207.