Continuous Reservoir Computing for Real-Time Locomotion Recognition in Wearable Robotics

Chang Zhao[1,2], Yuanyang Guo[1], Robin Degraeve[1], Jonas Eerdekens[1], Louis Flynn[2] and Bram Vanderborght[1,2]

[1] imec, Leuven, Belgium; [2] Brubotics, Vrije Universiteit Brussel (VUB), Brussels, Belgium

Reservoir Computing (RC) is attractive for temporal artificial intelligence because a fixed nonlinear dynamical system projects input streams into high-dimensional states while only a simple readout is trained [1]. This is well matched to wearable robotics, where locomotion mode recognition (LMR) must be accurate, stable, low-power, and fast. In prostheses and exoskeletons, LMR sits upstream of gait-phase estimation and torque generation; delayed mode recognition therefore limits how quickly the controller can react to stairs, ramps, and level walking [2]. Yet many data-driven LMR pipelines still rely on fixed trailing windows, so even a fast CNN remains constrained by a 0.5 s input-buffer latency floor after each transition.

We propose Continuous Reservoir Computing (CRC), a streaming LMR framework that replaces repeated window resets with persistent reservoir dynamics. CRC combines Phase Space Reconstruction (PSR) with a no-reset reservoir. PSR uses the current IMU sample and a delayed observation of the same signal to form a compact time-delay embedding inspired by Takens' theorem [3]. After quantization, the active phase-space cell generates a sparse pulse stream. At every 200 Hz sensor step, only the nodes associated with the active cell receive a pulse, while the remaining independent leaky-memory nodes relax. The reservoir state is carried forward across the full trial, and a lightweight Ridge readout maps it to locomotion scores at a 10 Hz control rate. Because temporal memory is supplied by PSR-driven fading-memory nodes rather than a dense recurrent matrix, CRC is compatible with compact physical-reservoir implementations [4].

CRC is evaluated on the public CAMARGO lower-limb locomotion dataset [5] using foot and thigh IMUs. Five classes are considered: level walking, stair ascent, stair descent, ramp ascent, and ramp descent. From 22 subjects, we reconstruct 2990 continuous evaluation sequences and apply leave-one-subject-out causal replay without look-ahead. We report time-weighted accuracy (TWA), time to confirm true transitions (TTC), false transition rate (FTR), and an explicit latency decomposition.

With a two-dimensional PSR encoder using $B = 13$ bins and delay $\tau = 7$ samples, and reservoir time constants $(\tau_a, \tau_r) = (14, 250)$, raw CRC achieves TWA $= 0.897$, TTC $= 0.258$ s, and FTR $= 31.1$/min under the 10 Hz causal protocol. Its structural algorithmic latency floor is only 35 ms, set by the PSR embedding horizon. Adding score-domain exponential moving average preserves TWA $= 0.897$, changes TTC only slightly to 0.272 s, and reduces FTR to 25.7/min without increasing this floor. The strongest compact window baseline, a dual-branch CNN with EMA, reaches a slightly higher TWA of 0.903 and faster TTC of 0.187 s, but remains pinned to a 0.5 s buffer floor. CRC also uses a much lighter decision head: 10,140 parameters and about 1.01×10^4 MACs per decision step, compared with 359,050 parameters and 3.99×10^6 MACs for the dual-fused CNN.

A targeted ablation confirms the source of this latency advantage. When the reservoir state is reset for each trailing window, performance drops to TWA $= 0.840$ and TTC $= 0.570$ s, and the method falls back into the same 0.5 s buffer-limited regime as window classifiers. Rather than optimizing all metrics uniformly, CRC opens a distinct low-latency operating regime with near-CNN recognition quality and a far smaller deployment footprint. The main remaining bottleneck is causal transition stabilization, motivating learned score-space filters and physical-reservoir implementation.

[1] G. Tanaka et al., Neural Netw., 115, 100-123, 2019.

[2] R. Gehlhar et al., Annu. Rev. Control, 55, 142-164, 2023.

[3] F. Takens, Lect. Notes Math., 898, 366-381, 1981.

[4] Y. Guo et al., Lect. Notes Comput. Sci., 15025, 156-167, 2024.

[5] J. Camargo et al., J. Biomech., 119, 110320, 2021.