

런타임 호출 주기 (grounding\_skip\_n=3, 실효 그라운드링 ≈1.3Hz)

서빙 제어 파라미터 요약

owlv2\_thresh — 코드 기본 0.25 / 운영 0.20 (VLA\_OWL2\_THRESH) · grounding\_skip\_n = 3

콜드스타트 가드 3스텝 (window=6에서 파생) · 회전 직후 강제 재검출 · 연속 회전 → 정지 대체 (상시)

제어 계층: 명령 COAST 4.0s·10Hz 재발행 / 회전만 move\_and\_stop\_ramped(0.4s) 후 자동정지

속도 크기는 사용하지 않음 — 펌웨어가 방향만 반영(상수 속도) · 상세 근거는 하단 각주

호출 1 — 전체 그라운드링



호출 2 — Kosmos-2만 (bbox 상속)



호출 3 — Kosmos-2만 (bbox 상속)



반복  
(호출1로)

예외 — 회전(ROT\_L/ROT\_R) 선택 시

CAN 명령이 COAST 4.0s·10Hz 반복 발행 대신 move\_and\_stop\_ramped(0.4s)로 짧게 실행 후 자동정지 —

동시에 skip\_n 무시하고 다음 호출에서 강제로 호출1(전체 그라운드링)을 수행. 직전도 회전이면 정지로 대체.

호출 1 지연 ≈ 1956ms (OWL-v2 1901.7ms가 97%) · 호출 2·3 지연은 Kosmos-2 비전만이라 훨씬 짧음

→ 3회 평균 실효 그라운드링 주기 ≈ 1.3Hz. bbox는 최대 2회(호출) 낮은 값을 사용

참고 — 학습(hold-aware)은 결정 간격 5프레임·다수결 라벨로 이 운영 주기를 반영해 설계됨(5.2절)

단 학습 윈도우 표집(간격 5)과 서빙 히스토리(연속 6개)의 시간 폭 일치는 프레임 타임스탬프로

직접 검증하지 않음 — 검증 항목으로 남김

근거: robovm\_nav/serve/stage2\_v2\_inference\_server.py(grounding\_skip\_n·just\_rotated) · robovm\_nav/serve/vla\_control\_utils.py(move\_duration=0.4) · ROS\_action/.../omni\_controller(Psd/Ultrasonic dev="can0")