

Reinforcement Learning for UR7e Trajectory Following and UR5e Manipulation

Hugo Simon, Dev Kumar, Basile Kulik, Kevin Belin

▶ **To cite this version:**

Hugo Simon, Dev Kumar, Basile Kulik, Kevin Belin. Reinforcement Learning for UR7e Trajectory Following and UR5e Manipulation. Independent research; original academic project at École Nationale Supérieure d'Arts et Métiers. 2026, pp.1-71. ⟨hal-05873384⟩

HAL Id: hal-05873384

https://hal.science/view/index/docid/5873384

Submitted on 20 Nov 2026

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Copyright - All rights reserved

Reinforcement Learning for UR7e Trajectory Following and UR5e Manipulation

Robot geometry, visual estimation, control
and reproducible simulation benchmarks

Hugo SIMON, Dev KUMAR, Basile KULIK and Kevin BELIN

École Nationale Supérieure d'Arts et Métiers

Paris, France

2025–2026 mechatronics project
Research synthesis and numerical audit · November 2026

Abstract

We study Cartesian target acquisition and laser line following on a six-axis UR7e system, with a related UR5e pick-and-place simulation. The work combines rigid-body and ray-plane geometry, visual tracking, Kalman estimation, damped inverse control and Soft Actor-Critic (SAC). We audit a frozen kinematic benchmark comprising 100 paired paths and three noise levels. Replaying the 600 public episodes on CPU reproduces all strict completion decisions and step counts. At zero noise, fixed damped least squares (DLS) has lower mean episode tracking RMS (2.38 versus 5.21 mm), while SAC reaches the 25 mm endpoint criterion on 93 cases versus 90 for DLS. The three discordant cases provide limited evidence of a completion advantage. The complete-model actor separately passes seven of ten offline wall cases with synthetic visual features. New numerical tests quantify calibration, spot visibility and filter sensitivity. The real camera detects the stationary laser and line. Loss of spot measurements in motion prevents repeatable end-to-end hardware validation.

Keywords Robot learning, SAC, DLS, UR7e, UR5e, visual servoing, laser trajectory, inverse kinematics, Kalman filtering, sim-to-real, hindsight replay.

Résumé

Le suivi de courbes sur UR7e est étudié par simulation et par intégration expérimentale. Le rejeu CPU des 600 épisodes reproduit les décisions et les nombres de pas. Sans bruit, DLS et SAC donnent respectivement 2,38 et 5,21 mm de RMS, avec 90 et 93 réussites sur 100 au seuil terminal de 25 mm. Le rapport couvre aussi la prise et dépose sur UR5e. La perte du laser en mouvement laisse le suivi réel en boucle fermée à valider.

Study record

The 2025–2026 project data and public software snapshots underlie the report. The November 2026 evaluation uses a frozen actor. The benchmark, offline replay, synthesis and archived project figures are identified by source and experimental scope.

Nomenclature

Notation	Meaning
UR7e / UR5e	Six-axis manipulator models used in separate experiments
FK (MGD), IK (MGI)	Forward and inverse geometric or kinematic mappings
TCP, B, F, W, C	Tool centre point; base, flange, wall and camera frames
q, $\dot{q}$, τ	Joint position, joint velocity and torque
n, d, ℓ, c	Wall normal, beam direction, intersection parameter and plane offset
J_v, J_W	TCP position Jacobian and wall-spot Jacobian
DLS, λ, σ_i	Damped least squares, damping and singular value

Notation	Meaning
H, KLT, HSV	Homography or measurement matrix, optical feature tracking and colour coordinates
KF, EKF, NIS, NEES	Filtering and normalised innovation/estimation error statistics
P, Q, R	Covariance of state, process and measurement within the filter section
SAC, PPO, HER	Soft Actor-Critic, Proximal Policy Optimization, Hindsight Experience Replay
α, γ, τ	Entropy temperature, discount and target-update factor in SAC
RMS, mm, mrad	Root mean square, millimetres and milliradians

Contents

Abstract 1
Résumé 1
Study record 1
Nomenclature 1
1. Research problem and evidence 3
1.1 Tasks and study questions 3
1.2 Provenance and comparable evidence 3
1.3 Experimental endpoints 4
2. Robot geometry and mechanical model 4
2.1 Platform and coordinate conventions 4
2.2 Direct kinematics 5
2.3 Geometric Jacobian and nominal inverse 6
2.4 Inertia, gravity and simulated contact 6
3. Wall task and automatic control 9
3.1 Intersection of the laser ray with the wall 9
3.2 Damped inverse, redundancy and limits 10
3.3 Classical optimal control and the implemented attenuation 11
3.4 Path parameter, closest point and task control 15
4. Camera geometry and visual tracking 15
4.1 Projective measurement of a planar target 15
4.2 Estimating and checking the homography 15
4.3 Line mask, laser candidates and local error 16
4.4 The Kanade–Lucas–Tomasi step 19
4.5 Controlled stress test of spot visibility 19
5. Fusion and observer consistency 20
5.1 Two measurement sources on one plane 20
5.2 Kalman update and camera acceptance 23
5.3 Numerical test with missing and contaminated images 24
6. Reinforcement learning formulations 29
6.1 State, history and reward 29
6.2 Soft Actor-Critic 29
6.3 Proximal Policy Optimization 30
6.4 Goal conditioning, HER and demonstrations 30
7. Preliminary experiments and target reaching 33
7.1 Two-axis inverted pendulum and planar arm 33
7.2 Reaching in a 180° sector 34
7.3 Full workspace, curricula and a fixed obstacle 35
8. The frozen 29-dimensional line follower 36
8.1 State and trajectory conventions 36
8.2 Rewards, progression and how early termination arises 37
8.3 Historical stress tests on shapes and lettering 38
9. Paired SAC versus DLS benchmark 38
9.1 Protocol and reproduction 38
9.2 Primary results and paired uncertainty 39
9.3 Shape, endpoint radius and performance trade-off 41
10. Full wall-following architecture and results 44
10.1 Closed loop, simulator and observation contract 44
10.2 Reward, ordered coverage and timing 45
10.3 The 520-row training record 46
10.4 Independent replay of the ten offline cases 46
11. Related UR5e pick-and-place simulation 48
11.1 Plant, policy and object goals 48
11.2 Sparse return and an executable relabelling example 48
11.3 Reading the manipulation curves 49
12. Real-robot integration and transfer limit 50
12.1 Coordinate registration and available observations 50
12.2 Timing and plausible failure mechanisms 51
12.3 Generalisation and experiment design 54
13. Discussion and conclusions 54
References 55
Appendix A. Procedure and model checks 57
A.1 Trajectory benchmark procedure 57
A.2 Numerical and observational checks 57
Appendix B. Original project figures 58
Appendix C. Data, software and reproduction 70

Theory boxes

Box 2.1 Screw kinematics and singular structure 7
Box 3.1 Damped least squares as regularisation 12
Box 3.2 Robust stability of the Cartesian servo 13
Box 4.1 Camera model and homography uncertainty 17
Box 4.2 Tracking limits and visual servoing 21
Box 5.1 Kalman filter: derivation and steady state 25
Box 5.2 Bias, correlation and robust fusion 27
Box 6.1 Foundations of the learning formulations 31
Box 9.1 Inference for paired comparisons 42
Box 12.1 Sim-to-real error budget and transfer 52

1. Research problem and evidence

1.1 Tasks and study questions

The mechatronics project conducted at École Nationale Supérieure d'Arts et Métiers in 2025–2026 studied how a six-axis collaborative manipulator can acquire a Cartesian path and follow it through reinforcement learning [1–5]. Its final objective was to trace a curve on a wall with a tool-mounted laser. The research also examined simpler problems that isolate individual difficulties: stabilising a two-direction cart-pendulum, inverting a planar two-link arm, reaching 3-D targets and passing a fixed obstacle. A related UR5e simulation explored grasping and placing a cube using goal-conditioned reinforcement learning [7].

The central engineering problem is the composition of four maps. Joint configuration determines the laser ray; ray and plane geometry determine the spot; a camera estimates the spot and the curve; a policy and a kinematic controller transform those estimates into motion. Errors in each map affect distinct quantities. A small nearest-path residual can coexist with a large endpoint error. Good forward kinematics on the nominal model can coexist with an incorrectly registered camera. Answering whether a learned policy helps therefore requires a fixed task definition, a comparable classical controller and metrics tied to the actual terminal state.

This report asks three questions. How do a frozen SAC policy and fixed damped least squares (DLS) compare when they receive the same target and perturbation? What do the archived simulation plots establish about perception, estimation and control? Which failure observed on hardware limits transfer, and how can its causes be separated? The first question is addressed with paired episode data and a fresh CPU replay. The second uses the executable source, the published training trace and the original plots. The third is bounded by the static observations and the team's two technical notices. Additional numerical experiments label their physical assumptions and use known synthetic truth.

1.2 Provenance and comparable evidence

The March report [1] documents the target-reaching comparison. The two June slidesets [2, 3] share several plots. The extended export adds the pick-and-place results and the complete laser architecture. The notices [4, 5] are distinct. The longer one adds the reaching and obstacle package and records the UR5e checkpoint provenance for the pick-and-place branch. The original project credits Hugo Simon, Dev Kumar, Basile Kulik and Kevin Belin. Code, notebook and model artifacts are cited separately at their accessible revisions.

The public UR7e repository [6] preserves separate simulation and real-implementation directories. Its simpler kinematic release [8] is a frozen, position-only experiment. The 600 episode rows on Hugging Face and Kaggle are byte-identical [8, 11]. The authors' 520-episode Monitor release [9] is a different run for the complete-model family. The Kaggle notebook is an analysis of the paired benchmark and the Policy Lab is an inference and visualisation port. They provide independent access to the same underlying experiment. The UR5e Kaggle notebook addresses sparse-reward relabelling with illustrative positions [13].

Table 1.1 – Task families and evidence used in the study.

Task	Policy and interface	Evidence available	Measured unit
CartPole and planar 2R	SAC/PPO and angle regression	Archived training and workspace plots	Episode duration, reachable coordinates
UR7e reaching and obstacle	SAC/PPO, HER, curriculum, IK variants	Archived results and source. External PyBullet robot package missing.	Endpoint distance, tolerance
Simple follower	SAC 29→6 and fixed DLS	600 paired releases and full CPU replay	RMS and strict terminal success
Full wall follower	SAC 33→2 plus geometry and filter	Monitor, offline weights and replay, Gazebo diagrams	Recent RMS and coverage
UR5e grasp and place	SAC/HER, 19+3+3 to 4	Simulation source, checkpoint and sparse-reward tutorial	Object-to-goal distance
Real UR7e	Camera, FK, observer, guarded policy	Static snapshot and moving-spot loss documented	Spot availability and residuals

The original presentation and subsequent repository updates were assembled over several design iterations. Accordingly, the stage, controller dimension, checkpoint and evidence source are stated with each quantitative result. Product ratings are distinguished from experimental values. The role of experimental reports is to describe what was observed, with reproducibility established separately for the public datasets and executable subset.

1.3 Experimental endpoints

For reaching, an episode's endpoint distance is $\|p_T - g\|_2$, and a success count depends on a declared tolerance. For line following, the acquisition flag concerns the first point, the progress variable records an ordered cursor or arc coordinate, an RMS error concerns the traversed path, and terminal distance concerns the final waypoint. The tolerance for each quantity and the samples used in its calculation are part of the metric.

For grasp and place, the simulated reward tests whether the object position enters a target ball. Quality after release adds a different requirement such as settled object pose and retention. For vision and fusion, metrics need a ground-truth spot, image age, spatial calibration and a way to account for missing detections. These endpoint definitions determine how the results can be compared across the project's tasks.

2. Robot geometry and mechanical model

2.1 Platform and coordinate conventions

The experimental hardware is a Universal Robots UR7e with six revolute joints and nominal reach 850 mm. The manufacturer specifies 7.5 kg payload, 20.6 kg mass including cable, IP54 arm enclosure, a maximum TCP speed of 4 m/s and pose repeatability ±0.03 mm under ISO 9283 [14]. Rated TCP speed and repeatability are characteristics of the product. Achievable absolute laser accuracy depends additionally on the laser origin and direction, camera calibration, wall geometry and the current robot configuration. The study uses separate nominal kinematic and factory-calibrated descriptions.

Write B for the robot base, F for the tool flange, L for the laser frame, C for the camera and W for the wall. The homogeneous transform ${}^{A}T_{B}$ maps frame B coordinates into frame A. Position coordinates use metres and joint angles use radians internally. The robot nominal tool frame, the simulated URDF end link and the geometrically calibrated beam direction may carry different constant rotations and offsets. A frame convention is specified alongside any position or Jacobian claim.



Figure 2.1 – Laboratory UR7e shown in the extended June presentation [3, slide 5]. The manufacturer ratings used in the text come from the datasheet [14].

2.2 Direct kinematics

For the values archived on the geometric modelling slide, the numerical transform is reproduced by the standard DH order $R_z(q_i)T_z(d_i)T_x(a_i)R_x(\alpha_i)$ [3, 15]. In this convention the revolute axis is the parent z-axis. The March report calls the convention “modified DH”. The product order and the numerical check below prescribe the convention actually used in this derivation. Craig’s modified convention changes which frame carries the fixed x rotation and displacement. A table can only be interpreted with its corresponding product.

$$A_i = R_z(q_i)T_z(d_i)T_x(a_i)R_x(\alpha_i), \qquad {}^{B}T_F = A_1A_2\cdots A_6. \tag{2.1}$$

Table 2.1 – Nominal standard DH parameters used in the geometric check [3].

Joint	a (m)	d (m)	α (rad)
1	0	0.1625	+π/2
2	−0.4250	0	0
3	−0.3922	0	0
4	0	0.1333	+π/2
5	0	0.0997	−π/2
6	0	0.0996	0

For clarity, applying R_z rotates the offset a_i into $(a_i\cos q_i, a_i\sin q_i)$. The orientation columns are obtained by postmultiplying $R_z\,R_x$. At all-zero joint angles, the archived translation comes from the two long links, the two offsets perpendicular to gravity and the difference in axial heights:

$$x_0 = a_2 + a_3 = -0.8172, \quad y_0 = -d_4 - d_6 = -0.2329, \quad z_0 = d_1 - d_5 = 0.0628. \tag{2.2}$$

The rotation at that pose is $R_x(\pi/2)$. For (0.3,−1.2,1,−0.8,1.4,0.5) rad, independent matrix multiplication gives position (−0.600753,−0.343087,0.665258) m. This pose and the archived zero transform are useful sign and unit checks. The presentation prints numerical inverse solutions at a reachable pose. Multiple solutions for a pose are expected for a 6R arm of this geometry. Their admissibility depends on limits, collision-free configurations and verification through the same forward map.

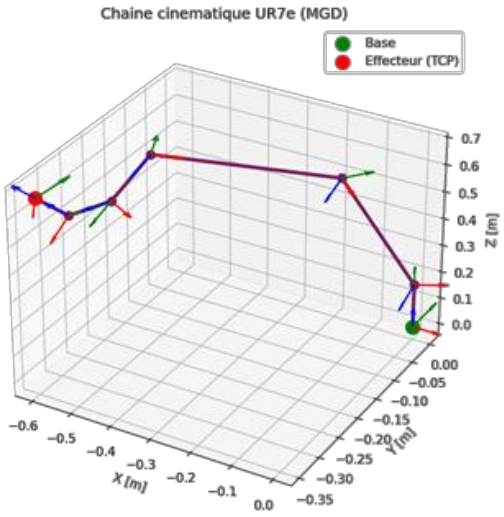


Figure 2.2 – Archived nominal configuration drawing from the geometric modelling slide [3, slide 14]. Local coordinates follow the conventions of that slide.

A second implementation, used in the frozen position-only release, runs forward kinematics through IKPy on the generated URDF. Its portable form uses static link transforms B_i followed by Rodrigues' formula for each joint rotation about unit axis a_i in its local frame:

$$\begin{aligned} R(a,\theta) &= I + \sin\theta [a]_\times \\ &\quad + (1-\cos\theta)[a]_\times^2, \\ T &= \prod_i B_i T(R(a_i, q_i)). \end{aligned} \tag{2.3}$$

Here $[a]_\times b = a \times b$. The archived chain matrices and axes are published with the release [8]. The matrices reflect the fixed link and end-frame conventions of that exporter. The numerical wall calculations in Sections 2 and 3 instead use the reported standard-DH parameters. Agreement within either convention is tested internally against its own finite differences.

2.3 Geometric Jacobian and nominal inverse

For joint i, let o_{i-1} and z_{i-1} be the origin and axis in frame B, evaluated at q. An infinitesimal rotation dq_i changes the tool position by $z_{i-1} \times (p - o_{i-1}) dq_i$. The columns of the linear and angular Jacobians are

$$\begin{aligned} J_{v,i} &= z_{i-1} \times (p - o_{i-1}), \\ J_{\omega,i} &= z_{i-1}, \\ \dot{p} &= J_v \dot{q}, \quad \omega = J_\omega \dot{q}. \end{aligned} \tag{2.4}$$

A geometric inverse searches q so that position and orientation errors vanish. A useful local rotation residual is the rotation-vector logarithm of $R_{des} R(q)^T$, expressed in B. Its small-angle direction matches the spatial angular Jacobian. A numerical step minimises a weighted residual with a damping term, followed by position, orientation and joint-limit tests. Rotational errors in radians can be mapped to equivalent lengths through a stated characteristic length. Different choices produce different condition numbers for a spatial 6×6 Jacobian.

For the shoulder chain, full-extension and folded elbow configurations near $\sin q_3 = 0$, wrist-axis alignment near $\sin q_5 = 0$, and shoulder alignment conditions are useful geometric indicators [1, 15]. The simple condition $x_5^2 + y_5^2 = 0$ in the initial notes is a reduced shoulder description. The offsets and reference definitions matter for a nominal UR arm. A singularity for a full pose can still have two independent directions on the wall. Conversely, ray-plane projection can become ill-conditioned before the pose Jacobian loses rank. The operational rank measure comes from the relevant task Jacobian, not only the named configurations.

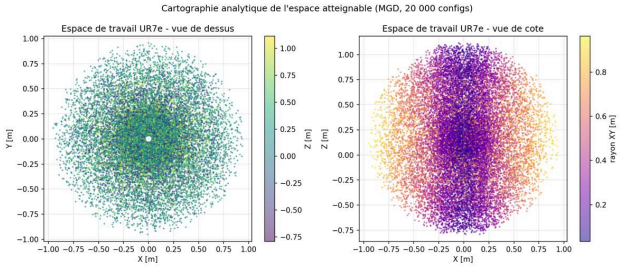


Figure 2.3 – Reach and nominal pose calculations from the geometric modelling slide [3, slide 14]. A sampled position set does not assert orientation or collision feasibility.

2.4 Inertia, gravity and simulated contact

An articulated rigid-body model follows from the link centres of mass, inertias and fixed frame transforms. Let m_i, I_i and R_i denote the mass, body inertia at the centre of mass and its orientation for link i. With J_{ci} and $J_{\omega i}$ the corresponding Jacobians, the kinetic energy gives

Box 2.1 Screw kinematics, singular structure and dynamic properties of the UR arm

Theoretical complement to Sections 2.1–2.4. Numerical values are computed by theory_ur7e.py (Appendix C) for the nominal DH model of Table 2.1.

Product of exponentials

The DH product (2.2) is one coordinate chart of the forward map. A frame-free alternative assigns to each revolute joint a unit axis ω_i through a point r_i, both expressed in B at $q = 0$, and the twist $\xi_i = (v_i, \omega_i)$ with $v_i = -\omega_i \times r_i$. With $M = T(0)$,

$$T(q) = e^{[\xi_1]q_1} e^{[\xi_2]q_2} \cdots e^{[\xi_6]q_6} M, \tag{B2.1}$$

where, for $\|\omega\| = 1$,

$$e^{[\xi]\theta} = \begin{bmatrix} e^{[\omega]\theta} & G(\theta)\, v \\ 0 & 1 \end{bmatrix},$$
$$G(\theta) = I\theta + (1 - \cos\theta)[\omega] + (\theta - \sin\theta)[\omega]^2, \tag{B2.2}$$

and $e^{[\omega]\theta}$ is Rodrigues' formula (2.3) [15]. The spatial Jacobian follows column by column from the adjoint map,

$$J_i^s = \mathrm{Ad}_{e^{[\xi_1]q_1} \cdots e^{[\xi_{i-1}]q_{i-1}}}\, \xi_i, \quad \mathrm{Ad}_T = \begin{bmatrix} R & [t]_\times R \\ 0 & R \end{bmatrix}, \tag{B2.3}$$

and the point Jacobian of (2.4) is recovered from $\dot{p} = v_s + \omega_s \times p$. Both charts describe the same manifold of poses; only the parameters exposed to calibration differ.

Closed-form singular structure

A change of reference point multiplies the 6×6 Jacobian by a unimodular block-triangular matrix, so $\det J$ is frame independent. For the UR geometry of Table 2.1 ($a_1 = a_4 = a_5 = a_6 = 0$) it factorises as

$$\det J(q) = a_2 a_3\, s_3\, s_5\, (a_2 c_2 + a_3 c_{23} + d_5\, s_{234}), \tag{B2.4}$$

with $s_{234} = \sin(q_2 + q_3 + q_4)$. The script checks (B2.4) against the numerical determinant on 2000 random configurations; the largest absolute difference is 9.7×10^{-17}. The three factors are the three singular families:

- elbow, $s_3 = 0$: upper arm and forearm aligned (stretched or folded);
- wrist, $s_5 = 0$: axes 4 and 6 parallel;
- shoulder, $a_2 c_2 + a_3 c_{23} + d_5 s_{234} = 0$: the intersection of axes 5 and 6 lies in the plane that contains axis 1 and is parallel to axis 2.

The zero configuration of Section 2.2 has $s_3 = s_5 = 0$ and is therefore doubly singular, which is why it serves only as a sign check. At the example pose $(0.3, -1.2, 1, -0.8, 1.4, 0.5)$ rad, $\det J = -0.0860\,\mathrm{m}^3$ and the shoulder term equals -0.622 m; at the HOME configuration of the frozen environment $\det J = 0.0820\,\mathrm{m}^3$.

Manipulability and conditioning

The image of the unit joint-velocity ball, $\{J_v \dot{q} : \|\dot{q}\| \le 1\}$, is an ellipsoid with semi-axes $\sigma_i u_i$ from $J_v = U \Sigma V^\mathsf{T}$. Its volume measure and anisotropy are

$$w = \sqrt{\det(J_v J_v^\mathsf{T})} = \sigma_1 \sigma_2 \sigma_3, \qquad \kappa = \sigma_1 / \sigma_3. \tag{B2.5}$$

The force ellipsoid of the same pose has semi-axes $1/\sigma_i$: directions that are fast in velocity are weak in force. For the position Jacobian,

Pose	$\sigma_1, \sigma_2, \sigma_3$ (m/rad)	w	κ
Example	0.991, 0.692, 0.195	0.134	5.07
HOME	0.723, 0.525, 0.302	0.115	2.39

These values fix the scale against which the damping λ of Section 3.2 must be compared (Box 3.1): with $\lambda = 0.02$ m/rad the damping is an order of magnitude below σ_3 at both poses.

For the full 6×6 Jacobian the rows mix m/rad and rad/rad, so κ depends on the characteristic length L of Section 2.3. With $J_L = \left[\begin{smallmatrix} J_v \\ L J_\omega \end{smallmatrix}\right]$, $L = 1$ m gives $\kappa(J_L) = 12.6$ at the example pose and 8.31 at HOME. Minimising over L gives $L^* = 0.336$ m with $\kappa = 7.93$, and $L^* = 0.360$ m with $\kappa = 5.04$. The optimum is close to the forearm length $|a_3| = 0.392$ m, the classical interpretation of a characteristic length.

Inverse geometry

With axes 2, 3 and 4 parallel, the UR chain admits a closed-form inverse. q_1 follows from the wrist-centre position (two shoulder branches), q_5

from the tool approach direction (two wrist branches) and q_3 from the law of cosines in the plane of axes 2–4 (two elbow branches). q_2, q_4 and q_6 are then unique. A regular pose therefore has at most $2^3 = 8$ solutions, which is the multiplicity noted in Section 2.2. Branches merge exactly on the singular sets of (B2.4). The numerical inverse of Section 2.3 minimises the weighted residual

$$e_L(q) = \begin{bmatrix} p_d - p(q) \\ L \log(R_d R(q)^\mathsf{T})^\vee \end{bmatrix} \tag{B2.6}$$

by Levenberg–Marquardt steps $\Delta q = (J_L^\mathsf{T} J_L + \mu I)^{-1} J_L^\mathsf{T} e_L$. The method is locally quadratic near a regular solution when $\mu \to 0$ and degrades gracefully, like DLS, near the singular sets.

Kinematic calibration as an identification problem

Write the tool point as $p = f(q, \theta)$, where θ stacks the DH parameters and the flange-to-laser transform. Around the nominal θ_0, $\delta p_k \approx \Phi(q_k)\,\delta\theta$ with $\Phi = \partial f / \partial \theta$. Stacking K poses gives the least-squares estimate

$$\widehat{\delta\theta} = (\Phi^\mathsf{T} W \Phi)^{-1} \Phi^\mathsf{T} W\, \delta p, \quad \mathrm{Cov}(\widehat{\delta\theta}) = (\Phi^\mathsf{T} W \Phi)^{-1}, \tag{B2.7}$$

with W the inverse measurement covariance. Parameters are identifiable only in the column space of Φ; the inverse condition number $\sigma_{\min}(\Phi)/\sigma_{\max}(\Phi)$ is the usual observability index used to choose calibration poses. The factory calibration extracted in Section 12.1 identifies the arm part of θ; the beam origin and direction remain unknown parameters of the same model.

Structural properties of the dynamic model

For (2.5)–(2.6), $M(q)$ is symmetric positive definite with $\lambda_{\min} I \le M(q) \le \lambda_{\max} I$. With the Christoffel construction of C, the matrix $N = \dot M - 2C$ is skew-symmetric, so $\dot q^\mathsf{T} N \dot q = 0$ and the total energy $E = \tfrac{1}{2}\dot q^\mathsf{T} M \dot q + V$ obeys

$$\dot E = \dot q^\mathsf{T}(\tau_{\mathrm{act}} - \tau_f) + \dot q^\mathsf{T} J_v^\mathsf{T} f_{\mathrm{ext}}. \tag{B2.8}$$

The map $\tau_{\mathrm{act}} \mapsto \dot q$ is therefore passive, the property behind passivity-based and impedance controllers. The model is also linear in the inertial parameters, $M\ddot q + C\dot q + \nabla_q V = Y(q, \dot q, \ddot q)\,\pi$ with $\pi \in \mathbb{R}^{10\times 6}$ (mass, first moments and inertia of each link). An added laser fixture changes only the parameters of link 6, which a persistently exciting trajectory and $\hat\pi = (Y^\mathsf{T} Y)^{-1} Y^\mathsf{T} \tau$ on the base-parameter subset would identify. No such identification was performed in the project; this explains why Section 2.4 assigns no torque margin to the laser trials.

By the principle of virtual work, a static tool load F_t applied at the TCP requires the joint torques

$$\tau = J_v^\mathsf{T} F_t, \qquad \|\tau\| \le \sigma_1 \|F_t\|. \tag{B2.9}$$

The most heavily loaded joint direction is the first right singular vector of J_v. At the example pose a 1 kg fixture ($\|F_t\| = 9.81$ N) bounds the induced torque norm by 9.7 N m, before link and cable masses are added.

Task dimension and redundancy

Let $x = f(q) \in \mathbb{R}^m$ be the task variable. Where rank $J = m$, the implicit function theorem makes the preimage $\{q : f(q) = x\}$ a smooth self-motion manifold of dimension $r = 6 - m$. The three tasks of the report therefore differ in kind. The full tool pose has $m = 6$ and $r = 0$, so only isolated solutions exist. The 3-D position benchmark of Section 8 has $m = 3$ and $r = 3$. The laser spot on the wall has $m = 2$ and $r = 4$, since the spot depends on the beam line and not on the rotation about it nor on the travel along it. The wall Jacobian $J_W = B_W P_d (J_o - \ell [d]_\times J_\omega)$ of (3.2) is 2×6, so its null space has dimension at least 4 everywhere. The secondary motion of Section 3.2 can then pursue up to four additional objectives, such as joint centring, manipulability or incidence angle, without moving the spot to first order. Rank loss of J_W has two distinct origins: an arm singularity that removes beam motions, or the projector P_d, whose norm $1/|n^\mathsf{T} d|$ grows at grazing incidence.

Second-order differential kinematics

Differentiating $\dot p = J_v \dot q$ gives

$$\ddot p = J_v \ddot q + \dot J_v \dot q, \qquad \dot J_v \dot q = \sum_{k=1}^{6} \frac{\partial J_v}{\partial q_k} \dot q_k\, \dot q, \tag{B2.10}$$

with $\partial J_{v,i} / \partial q_k = z_{k-1} \times J_{v,i}$ for $k \le i$, and $z_{i-1} \times J_{v,k}$ for $k > i$. The quadratic velocity term is what a first-order (velocity-level) wall controller neglects. It scales with $\|\dot q\|^2$ and is small at the 0.35 rad/s joint-speed cap of Section 10.2.

$$T_{\text{kin}} = \frac{1}{2}\dot{q}^{T} M(q)\dot{q},$$
$$M = \sum_i m_i J_{c_i}^{T} J_{c_i} + \sum_i J_{\omega_i}^{T} R_i I_i R_i^{T} J_{\omega_i}. \tag{2.5}$$

Taking $V=\sum_i m_i g z_{ci}$, the Euler–Lagrange equation yields

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + \nabla_q V + \tau_f = \tau_{\text{act}} + J_v^{T} f_{\text{ext}}. \tag{2.6}$$

A consistent Coriolis matrix may be built from Christoffel terms $C_{ij}=\sum_k \Gamma_{ijk}\dot{q}_k$, with $\Gamma_{ijk}=(\partial_k M_{ij}+\partial_j M_{ik}-\partial_i M_{jk})/2$.
A full wrench also includes $J_\omega^{T}\tau_{\text{ext}}$. This model separates inertial effects, friction and external loads.

URDF specifies inertial and collision elements for the simulated structure, and Gazebo or PyBullet integrates the selected dynamics. The simplified frozen benchmark only updates joint positions and evaluates FK. It therefore measures the algorithmic quality of a kinematic trajectory and omits motor torques, compliance, contact and time-dependent velocity response.

In the laboratory, a beam mounted on the tool has both an offset and an orientation relative to F. Added tool and cable masses change the inertia and permitted payload operating envelope. These effects are configuration-dependent and require the actual payload and centre of mass. The report assigns no measured torque or force margin to the laser trials.

3. Wall task and automatic control

3.1 Intersection of the laser ray with the wall

Let the laser origin be o(q) and its direction d(q), a unit vector in B. With wall normal n and offset c, the plane satisfies $n^{T}p=c$. The positive intersection is

$$\ell(q) = \frac{c - n^{T} o(q)}{n^{T} d(q)}, \qquad p_W(q) = o(q) + \ell(q) d(q). \tag{3.1}$$

This map requires $n^{T}d\neq 0$ and $\ell>0$. A valid point for following additionally lies within the wall polygon. The simulation uses a wall normal aligned with the base X-axis. The projectors along its Y and Z directions form the two rows of a matrix B_W, and a fixed wall origin p_0 gives $s=B_W(p_W-p_0)$.

Differentiation of $n^{T}p_W=c$ with fixed plane gives $n^{T}\dot{p}_W=0$. Substituting $\dot{p}_W=\dot{o}+\dot{\ell}d+\ell\dot{d}$ shows that $\dot{\ell}=-n^{T}(\dot{o}+\ell\dot{d})/(n^{T}d)$. Angular velocity gives $\dot{d}=\omega\times d=-[d]_\times J_\omega\dot{q}$. The spot Jacobian is therefore

$$P_d = I - \frac{d n^{T}}{n^{T} d}, \qquad J_W = B_W P_d (J_o - \ell [d]_\times J_\omega), \qquad \dot{s} = J_W \dot{q}. \tag{3.2}$$

J_o includes the flange-to-laser offset. P_d maps vectors into the tangent plane along the ray, with an operator norm that grows at grazing incidence. The reference code numerically differentiates the laser intersection while guarding degenerate cases [6]. In the declared DH model, analysis at 40 random configurations with central step 10^{-5} rad gives a maximum elementwise FK Jacobian discrepancy of 2.01×10^{-11} m/rad. In the 34 of those configurations with $|n^{T}d|\geq 0.2$, a synthetic wall was placed 0.45 m along the ray separately for each configuration. The wall Jacobian differs from its finite difference by at most 6.15×10^{-9} m/rad. This checks algebra and implementation for a specified nominal model.

At fixed positive ray travel ℓ, a small tilt in the incidence plane causes first-order displacement of magnitude $\ell\,|\delta\theta|/|\cos\psi|$, with ψ the unsigned incidence angle to the normal. A normal plane offset δc changes the spot coordinates along the plane by $|\delta c|\tan\psi$. A 0.1° beam tilt at 0.5 m is therefore about 0.873 mm at normal incidence and 1.75 mm at 60°. A 1 mm disparity in the plane offset contributes another 1.73 mm at 60° when considered separately. The signs and directions govern whether errors cancel or add.

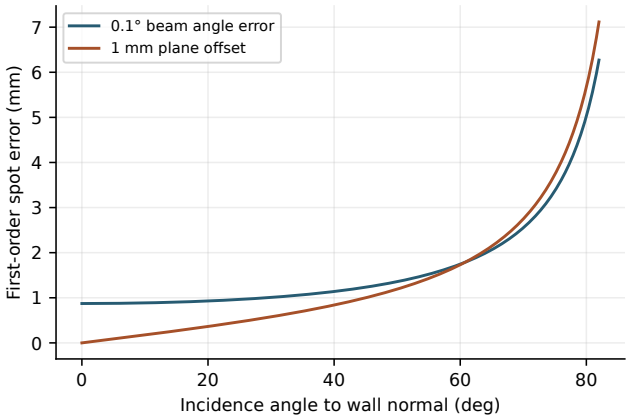


Figure 3.1 – First-order geometric sensitivity for a fixed nominal travel of 0.5 m. The calculation is independent of camera or joint noise.

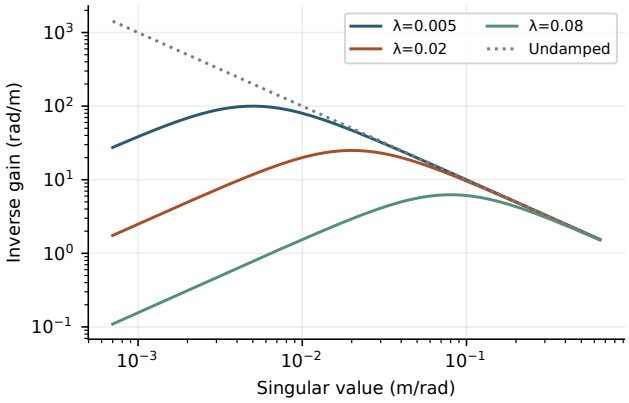


Figure 3.2 – Singular-direction inverse gain for several damping values. The frozen 3-D position baseline uses 0.02; the complete wall controller schedules its own damping.

3.2 Damped inverse, redundancy and limits

Given wall velocity v, define a least-squares task cost plus regularisation. Differentiating the cost gives the KKT equation for its unconstrained minimiser:

$$\begin{aligned} \dot{q}_\lambda &= \arg\min_{\dot{q}} \\ &\{\|J_W\dot{q} - v\|_2^2 + \lambda^2\|\dot{q}\|_2^2\}, \\ \dot{q}_\lambda &= J_W^T \\ &(J_W J_W^T + \lambda^2 I)^{-1} v. \end{aligned} \tag{3.3}$$

The singular-vector form follows from $J_W=U\Sigma V^T$. Each inverse gain is $\sigma_i/(\sigma_i^2+\lambda^2)$, bounded by $1/(2\lambda)$ for $\lambda>0$. The achieved velocity in the task direction is attenuated by $\sigma_i^2/(\sigma_i^2+\lambda^2)$. Regularisation therefore exchanges tracking bandwidth for a limit on joint amplification. Here λ has the same units as a singular value of J_W [16].

The full pipeline uses the minimum wall singular value with a nominal threshold 0.05 and schedules λ between 10^{-4} and 0.08 in the checked implementation. The Yoshikawa index $w=\sqrt{(\det(J_W J_W^T))}=\sigma_1\sigma_2$ depends on task units and both relative gains [17]. A determinant can conceal a weak direction compensated by a large one. Monitoring the singular values, w, and the ray denominator covers different failure modes.

The exact pseudoinverse gives the secondary motion $\dot{q}_N=(I-J_W^+J_W)\,k\nabla_q w$. It cancels first-order motion in the primary task. The code uses a damped inverse and clips or scales velocity. The identity in SVD coordinates is

$$\begin{aligned} &J_W(I - J_\lambda^\dagger J_W) \\ &= U\,\mathrm{diag}(\chi_i)V_r^T, \\ &\chi_i = \frac{\lambda^2\sigma_i}{\sigma_i^2 + \lambda^2}. \end{aligned} \tag{3.4}$$

Thus secondary motion can introduce finite wall velocity. Componentwise limits can also change the direction of a combined command. Uniform scaling of the final vector preserves its joint-space direction and scales the resulting linearised wall speed. It cannot undo leakage already introduced by damping or a diagonal correction. Local redundancy control and collision safety remain separate problems.

3.3 Classical optimal control and the implemented attenuation

For a linear discrete plant $x_{k+1}=Ax_k+Bu_k$ with cost $\sum (x^TQx+u^TRu)$, dynamic programming with quadratic value x^TPx yields

$$K=(R+B^TPB)^{-1}B^TPA, \quad u=-Kx. \tag{3.5}$$

Substitution into the value recursion gives the discrete algebraic Riccati equation:

$$P=Q+A^TPA - A^TPB(R+B^TPB)^{-1}B^TPA. \tag{3.6}$$

Under stabilisability and detectability, this defines an infinite-horizon state-feedback controller for the stated linear model [18]. It requires specified states, dynamics, control inputs and weights. For a scalar continuous first-order actuator $\dot{x}=-x+u$ the continuous Riccati equation $0=Q-2P-P^2/R$ gives $K=P/R=\sqrt{(1+Q/R)}-1$. This algebraic form explains the analogy used in the code.

The retained complete-model routine computes squared participation of each joint in the weakest right singular vector of J_W, normalises it by its maximum, and decreases the effective scalar R_i toward 0.05 from 1. It reports $K_i=\sqrt{(1+Q/R_i)}-1$ with $Q=1$, and applies $\dot{q}_{f,i}=\dot{q}_{d,i}/(1+K_i)$ with limits. Normalised gain varies from $1/\sqrt{2}=0.707$ for an unweighted axis to $1/\sqrt{21}=0.218$ at the strongest attenuation. The operation is a gain-scheduled velocity filter inspired by a scalar LQR. The program has no online six-axis Riccati feedback loop. Its plots of joint norms are evidence of command attenuation. Validating closed-loop stability would require the realised arm, timing, observation and saturation model.

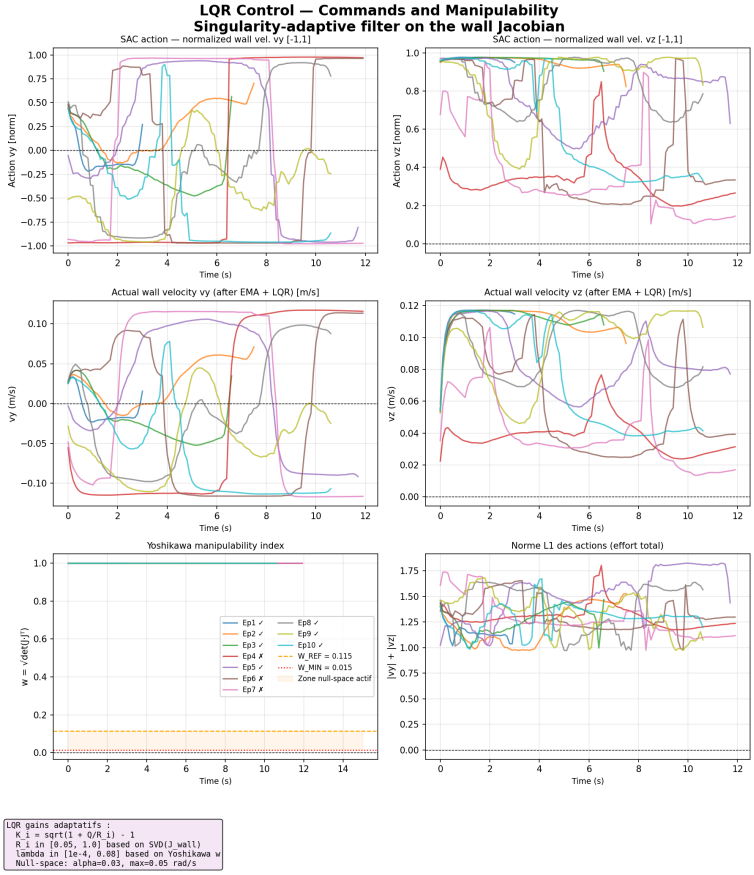


Figure 3.3 – Original diagnostic of joint commands and singularity-related attenuation from the complete-model code [6]. The offline generator supplies kinematic traces.

Box 3.1 Damped least squares as a regularised inverse problem

Theoretical complement to Section 3.2. HOME-pose singular values from Box 2.1; values computed by theory_ur7e.py.

Variational form and filter factors

The DLS command of (3.3) is the Tikhonov solution

$$\dot{q}_\lambda = \arg\min_{\dot{q}} \ \|J\dot{q} - v\|_2^2 + \lambda^2 \|\dot{q}\|_2^2. \tag{B3.1}$$

Setting the gradient to zero gives $(J^\mathsf{T}J + \lambda^2 I)\dot{q} = J^\mathsf{T}v$. The push-through identity $(J^\mathsf{T}J + \lambda^2 I)^{-1}J^\mathsf{T} = J^\mathsf{T}(JJ^\mathsf{T} + \lambda^2 I)^{-1}$ yields the two equivalent forms used in code. With $J = \sum_i \sigma_i u_i v_i^\mathsf{T}$,

$$\dot{q}_\lambda = \sum_i f_i \, \frac{u_i^\mathsf{T} v}{\sigma_i} \, v_i, \qquad f_i = \frac{\sigma_i^2}{\sigma_i^2 + \lambda^2} \in (0, 1). \tag{B3.2}$$

The filter factors f_i leave strong directions almost untouched and suppress weak ones. The task-space residual of the regularised step is a bias,

$$v - J\dot{q}_\lambda = \sum_i \frac{\lambda^2}{\sigma_i^2 + \lambda^2} \, (u_i^\mathsf{T} v) \, u_i, \tag{B3.3}$$

which vanishes only for $\lambda \to 0$.

Bias–variance trade-off

Let the requested velocity carry noise, $v = v^* + \eta$ with $\mathrm{Cov}(\eta) = s^2 I$, and let $\dot{q}^* = J^+ v^*$. Then

$$\mathbb{E}\|\dot{q}_\lambda - \dot{q}^*\|^2 = \sum_i (1 - f_i)^2 \frac{(u_i^\mathsf{T} v^*)^2}{\sigma_i^2} + s^2 \sum_i \frac{f_i^2}{\sigma_i^2}. \tag{B3.4}$$

The first term grows with λ and the second decreases with it. Each term of the noise sum is bounded by $f_i^2/\sigma_i^2 \le 1/(4\lambda^2)$, which is the gain bound $1/(2\lambda)$ of Section 3.2. Classical choices of λ follow from (B3.4): the discrepancy principle and generalised cross-validation. Variable damping that grows as a singular value or the manipulability falls below a threshold,

$$\lambda^2 = \begin{cases} 0, & w \ge w_0, \\ \lambda_0^2 \, (1 - w/w_0)^2, & w < w_0, \end{cases} \tag{B3.5}$$

is the singularity-robust schedule of Nakamura and Hanafusa [16]. The wall controller's threshold on $\sigma_{\min}$ (0.05) with $\lambda \in [10^{-4}, 0.08]$ belongs to this family.

Magnitude in the frozen benchmark

At HOME, $\sigma(J_v) = (0.723, 0.525, 0.302)$ m/rad. With $\lambda = 0.02$ the filter factors are $f = (0.9992, 0.9986, 0.9956)$, and the largest damped inverse gain is $\|J_\lambda^\dagger\| = 3.29$ rad/m against the worst-case bound of 25 rad/m. In that region the fixed comparator behaves as an almost exact pseudo-inverse; the damping matters only near the singular sets of Box 2.1.

Null space, leakage and gradient projection

For full row rank, $P_N = I - J^+J$ is the orthogonal projector onto $\ker J$, so $JP_N = 0$ and a secondary command $\dot{q}_N = P_N z$ produces no first-order task motion. With damping the projector is replaced by $I - J_\lambda^\dagger J$, and (3.4) shows a leakage

$$\|J(I - J_\lambda^\dagger J)\| = \max_i \frac{\lambda^2 \sigma_i}{\sigma_i^2 + \lambda^2} \le \frac{\lambda}{2}, \tag{B3.6}$$

with the maximum reached at $\sigma_i = \lambda$. For $\lambda = 0.02$ a unit secondary joint velocity leaks at most 10 mm/s into the wall plane. To increase manipulability one takes $z = k\nabla_q w$ with

$$\frac{\partial w}{\partial q_j} = w \ \mathrm{tr}\Big(J^+ \frac{\partial J}{\partial q_j}\Big), \tag{B3.7}$$

which follows from Jacobi's formula for $\partial \det(JJ^\mathsf{T})/\partial q_j$. For the wall task the null space has dimension 4 (Box 2.1), so this objective does not compete with spot tracking.

Limits as constraints

Joint-velocity limits turn (B3.1) into a bound-constrained quadratic programme,

$$\min_{\dot{q}} \ \tfrac{1}{2}\|J\dot{q} - v\|^2 + \tfrac{1}{2}\lambda^2\|\dot{q}\|^2 \quad \text{s.t.} \quad \underline{\dot{q}} \le \dot{q} \le \overline{\dot{q}}. \tag{B3.8}$$

At the optimum, the KKT conditions give $(J^\mathsf{T}J + \lambda^2 I)\dot{q} - J^\mathsf{T}v + \mu_+ - \mu_- = 0$, with complementary multipliers $\mu_\pm \ge 0$ on the active bounds. Uniform scaling of the unconstrained step is optimal only when the scaled vector already satisfies these conditions. Componentwise clipping, as in the benchmark ($|\Delta q_j/0.05| \le 1$), is generally not optimal: the same convex programme, solved exactly, returns the closest achievable task motion.

Box 3.2 Robust stability and noise propagation of the discrete Cartesian servo

Theoretical complement to Sections 3.2–3.3 and 9.2. All stability limits are verified numerically by root computation in theory_ur7e.py.

Closed-loop error model

Consider one control period of the fixed comparator (9.1) with a constant target g and an observation $p_{\text{obs}} = p + n_k$. With $e_k = g - p_k$ and the linearisation $\Delta p = J\Delta q$,

$$e_{k+1} = (I - \alpha J\hat{J}_\lambda^\dagger)\, e_k + \alpha J\hat{J}_\lambda^\dagger\, n_k, \qquad \alpha = 0.8, \quad \text{(B3.9)}$$

where $\hat{J}$ is the Jacobian used by the controller. With an exact Jacobian, $J\hat{J}_\lambda^\dagger = U\,\mathrm{diag}(f_i)U^\mathsf{T}$ and the modes decouple as $e_{k+1}^{(i)} = (1 - \alpha f_i)e_k^{(i)}$. The Lyapunov function $V = \|e\|^2$ satisfies $V_{k+1} \le \rho^2 V_k$ with $\rho = \max_i |1 - \alpha f_i| = 0.203$ at HOME. Each period therefore removes about 80% of the error.

Robustness to Jacobian error

Let $J = \hat{J} + E$, where E collects finite-difference error, joint-angle noise in $\hat{J}$ and model error. By the triangle inequality,

$$\|I - \alpha J\hat{J}_\lambda^\dagger\| \le \rho + \alpha\|E\|\,\|\hat{J}_\lambda^\dagger\|, \qquad \text{(B3.10)}$$

so the loop remains a contraction whenever

$$\|E\| < \frac{1-\rho}{\alpha\,\|\hat{J}_\lambda^\dagger\|} = 0.302 \text{ m/rad at HOME.} \qquad \text{(B3.11)}$$

This small-gain margin equals σ_3 itself. The Jacobian errors present in the benchmark are orders of magnitude smaller: the finite-difference error is about 2×10^{-11} m/rad (Section 3.1), and evaluating $\hat{J}$ at a joint angle perturbed by 5 mrad changes it by roughly $\|\partial J/\partial q\| \times 0.005 \approx 5 \times 10^{-3}$ m/rad. Noise therefore degrades the DLS baseline through the error signal, not through the Jacobian.

Propagation of observation noise

For one mode with $f_i \simeq 1$ and white noise of variance s^2, (B3.9) is a first-order autoregression whose stationary variance is

$$\mathrm{Var}(e) = \frac{\alpha^2 s^2}{1-(1-\alpha)^2} = \frac{\alpha}{2-\alpha}\, s^2 = 0.667\, s^2. \qquad \text{(B3.12)}$$

The benchmark metric is the distance to the nearest vertex. Noise along the path is partly absorbed by the nearest-vertex search, so the added mean-square error should lie between a lateral-only (two-axis) and an isotropic (three-axis) projection of (B3.12). From Table 9.1, the DLS mean-square RMS rises by $3.554^2 - 2.376^2 = 6.99\,\text{mm}^2$ at $\sigma = 2$ mm and by $36.6\,\text{mm}^2$ at $\sigma = 5$ mm. The model brackets are $[5.33, 8.00]$ and $[33.3, 50.0]\,\text{mm}^2$. Both measurements fall inside them, so the fixed gain explains the noise sensitivity of DLS reported in Section 9.2. A gain α also sets the steady lag behind a target advancing by δ per period, $e_{\text{ss}} = \delta/\alpha$. This gives the design trade-off

$$\mathrm{MSE}(\alpha) \approx \frac{\delta^2}{\alpha^2} + m\,\frac{\alpha}{2-\alpha}\, s^2, \qquad m \in \{2, 3\}, \qquad \text{(B3.13)}$$

whose minimiser moves to smaller gains as the sensor noise s grows.

Feedback delay

If the observation used at period k is d periods old, as for a camera spot with latency, each mode obeys

$$e_{k+1} = e_k - g\, e_{k-d}, \qquad g = \alpha f_i, \qquad \text{(B3.14)}$$

with characteristic polynomial $z^{d+1} - z^d + g$. The recursion is asymptotically stable if and only if

$$0 < g < g^*(d) = 2\sin\frac{\pi}{2(2d+1)}. \qquad \text{(B3.15)}$$

The boundary is reached when a root crosses the unit circle at $z = e^{i\varphi}$ with $\varphi = \pi/(2d+1)$. The script confirms (B3.15) to 10^{-15} for $d \le 5$, giving $g^* = 2,\ 1,\ 0.618,\ 0.445,\ 0.347$. For $g = 0.8$ the spectral radius is 0.20 at $d = 0$, 0.894 at $d = 1$ (slow, oscillatory) and 1.078 at $d = 2$ (unstable).

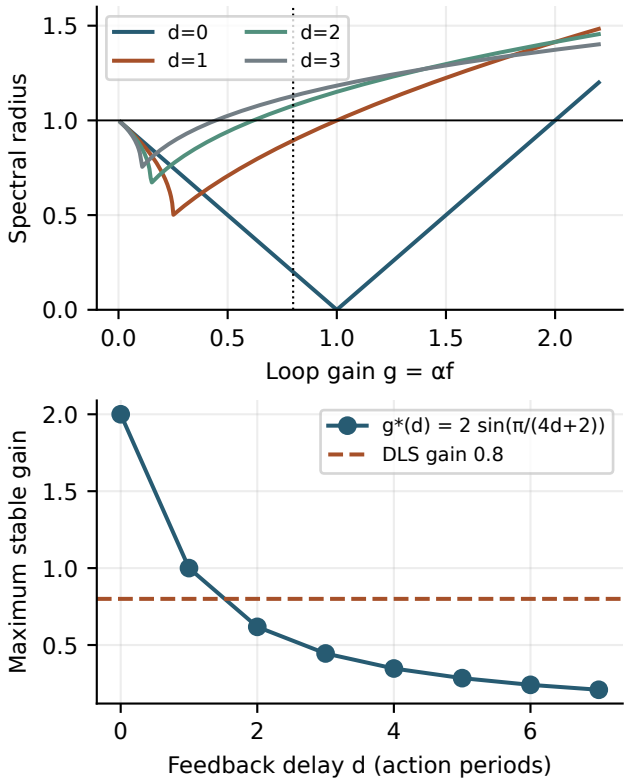


Figure B3.1 – Spectral radius of (B3.14) versus loop gain for $d = 0$–3 (top) and maximum stable gain $g^*(d)$ (bottom). The dotted and dashed lines mark the DLS gain 0.8.

With the 0.10 s action period of Section 5.1, the 80 ms timing mismatch of Section 12.2 is close to one period. A camera-closed version of the comparator would then sit near the poorly damped $d = 1$ case, and two periods of latency would destabilise it without a gain reduction or a prediction of the measurement to the actuation time.

Sensitivity and modulus margin

The loop transfer of one mode is $L(z) = g\,z^{-d}/(z-1)$ and the sensitivity is $S = 1/(1+L)$. The modulus margin $1/\|S\|_\infty$ is the smallest distance from the Nyquist locus to -1. For $g = 0.8$, $\|S\|_\infty = 1.67$ at $d = 0$ and 5.67 at $d = 1$, so the margin falls from 0.60 to 0.18. By the small-gain theorem, a multiplicative uncertainty Δ on the plant preserves stability if $\|\Delta\, T\|_\infty < 1$ with $T = 1 - S$. A delayed loop therefore tolerates much less unmodelled dynamics.

Action smoothing and actuator saturation

The classical wall servo of Section 3.4, $v_{\text{raw}} = c\,e$, combined with the smoothing $v_k = (v_{k-1} + v_{\text{raw}})/2$ of Section 10.2, gives with $k = cT$ the closed loop

$$z^2 - (1.5 - 0.5k)\,z + 0.5 = 0. \tag{B3.16}$$

Jury's conditions $P(1) = 0.5k > 0$, $P(-1) = 3 - 0.5k > 0$ and $|0.5| < 1$ give stability for $0 < k < 6$, which the script confirms. The smoothing raises the gain limit (2 without filter), but the product of the roots is fixed at 0.5. Their modulus therefore cannot fall below $\sqrt{0.5} = 0.707$: complex roots of exactly that modulus appear for $3 - 2\sqrt{2} < k < 3 + 2\sqrt{2}$. The filter thus bounds the fastest decay at 29% of the error per period, whatever the gain.

The velocity cap of Section 10.2 makes the loop nonlinear, $v = \text{sat}(v_{\text{raw}})$, with the saturation in the sector $[0, 1]$. By the discrete circle criterion, the loop is absolutely stable for every sector nonlinearity if

$$\min_{\omega\in[0,\pi]} \operatorname{Re}\{1 + k\,L_1(e^{\mathrm{j}\omega})\} > 0, \qquad L_1(z) = \frac{0.5\,z}{(z-1)(z-0.5)}. \tag{B3.17}$$

Numerically this holds for $k < 2/3$. The circle criterion is only sufficient, so the true limit lies between $2/3$ and the linear bound 6. Gains above $2/3$ need a saturation-aware argument, such as anti-windup or a Lyapunov function of the saturated loop.

Guaranteed margins of the LQR of Section 3.3

For the scalar actuator $\dot{x} = -x + u$ with cost weights $Q = 1$ and R, the loop $L(s) = K/(s+1)$ with $K = \sqrt{1 + Q/R} - 1$ satisfies Kalman's return-difference equality,

$$|1 + L(\mathrm{j}\omega)|^2 = 1 + \frac{Q}{R}\,\frac{1}{1+\omega^2} \;\geq\; 1. \tag{B3.18}$$

The Nyquist locus therefore avoids the unit disc centred at -1. This guarantees a gain margin $[\tfrac{1}{2}, \infty)$ and a phase margin of at least 60°. For $R = 1$, $K = 0.414$ and $|L| < 1$ at all frequencies (infinite phase margin). For $R = 0.05$, $K = 3.583$, the crossover is 3.44 rad/s and the phase margin is 106.2°. These guarantees hold for continuous-time state feedback only. Discrete-time LQR has no such universal margins, and the routine of Section 3.3 uses the formula as a static attenuation, not as a feedback loop, so no closed-loop margin can be claimed for it.

3.4 Path parameter, closest point and task control

For an ordered polyline with vertices p_i, the closest point on a segment is $p_i+\beta(p_{i+1}-p_i)$, with β the clipped scalar projection. Accumulating the segment lengths before that point defines a curvilinear coordinate. A local tangent $t=(p_{i+1}-p_i)/\|p_{i+1}-p_i\|$ and perpendicular unit vector ν resolve tangential progress $t^T v$ and lateral tracking error $e_{perp}=\nu^T(s-p_{proj})$. At crossings, global nearest-point selection can jump in arc length. An ordered search window and bounded progress step maintain continuity.

A classical wall servo could use $v=v_t\, t-k_e\, e_{perp}\nu$, with arc-speed and endpoint rules. A step of DLS minimises the residual towards that commanded velocity. The frozen comparator analysed in Section 9 is a different baseline. It receives the current 3-D target point, no wall-plane tangent, and sends a damped joint increment derived from that point. Both algorithms in that benchmark use the same target sequence and motion cap. An endpoint servo or an MPC baseline is a separate controller choice to be evaluated with its own fixed parameters.

4. Camera geometry and visual tracking

4.1 Projective measurement of a planar target

In an undistorted pinhole image, a 3-D point P with camera coordinates (X_C,Y_C,Z_C), $Z_C>0$, projects to $u=f_xX_C/Z_C+c_x$, $v=f_yY_C/Z_C+c_y$. The intrinsics are defined for the actual resized stream. If $P=p_0+b_y\, y+b_z\, z$ is on the wall, ${}^{C}T_B=(R_C,t_C)$ supplies a planar projectivity:

$$\begin{aligned} \tilde{u} &\propto H_{W\to I}\tilde{s}, \\ H_{W\to I} &= K[R_C b_y \;\; R_C b_z \;\; R_C p_0 + t_C], \\ \tilde{s} &= (y, z, 1)^T. \end{aligned} \tag{4.1}$$

After lens distortion correction, $H_{I\to W}=H_{W\to I}^{-1}$ maps an image ray back to the wall, up to scale. Image position alone carries no depth when the spot leaves the calibrated wall. For any homography h defined by its rows h_1,h_2,h_3, the map is $(h_1^T\tilde{u},h_2^T\tilde{u})/(h_3^T\tilde{u})$. Differentiation gives Jacobian entries $(h_{ij}-s_i\, h_{3j})/(h_3^T\tilde{u})$ for the two image coordinates. This spatially varying Jacobian converts image covariance to metric covariance. Uncertainty in H, target location and lens distortion adds to the propagated pixel term.

At a nominal 320 by 240 resized image and 75° horizontal FOV in the simulator, $f_x=320/[2\tan(75°/2)]=208.5$ px. At 1.2 m and fronto-parallel incidence the local scale is 5.75 mm/px. It varies across the plane in an oblique view. Camera position and nominal angle recorded in the slide therefore belong to the simulated setup, with no inference about real calibration error.

4.2 Estimating and checking the homography

A planar calibration correspondence yields two homogeneous linear constraints: $h_1^T\tilde{u}-y\, h_3^T\tilde{u}=0$ and $h_2^T\tilde{u}-z\, h_3^T\tilde{u}=0$. Stack them, centre and scale both point sets to mean distance $\sqrt{2}$, and select the right singular vector for the smallest singular value [19]. Four non-collinear and appropriately spread pairs can identify eight degrees of freedom. Additional points overdetermine the problem. Checking errors only at fitted points conceals overfitting and coverage gaps. For a full calibration, undistort with the camera model and assess reprojection and metric residuals at independent locations [20].

A synthetic comparison sets a 0.60 by 0.40 m wall patch at 1.2 m with orientation from 28° and −12° rotations. Pixel coordinates of four corners or 25 grid fiducials receive iid 0.65 px component noise. Each trial estimates pixel-to-wall H with normalised DLT and evaluates 315 noise-free interior points. Across 500 trials per design, four corners give a median RMS of 4.14 mm (95th percentile 6.94 mm), and the 25-point design gives 1.90 mm (2.86 mm). The input is a pinhole model with known correspondences, fixed camera and no distortion. Comparison separates consequences of

point count and spatial coverage from unrelated hand–eye fitting errors.

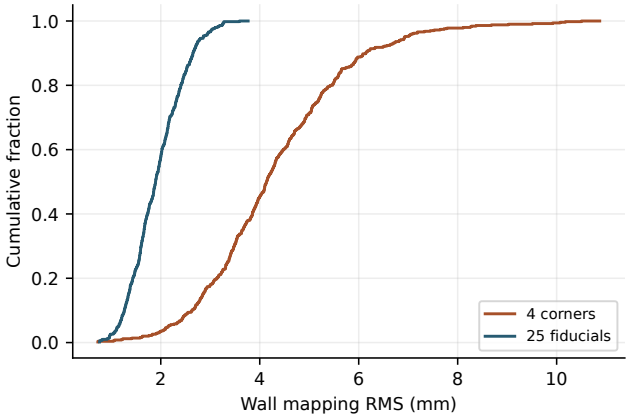


Figure 4.1 – Synthetic calibration with a fixed oblique pinhole camera and independent fiducial localisation noise. Error is evaluated on a shared interior test grid.

When the camera is calibrated by moving a laser to reference spots estimated through FK, any FK registration bias propagates into the inferred homography. A residual comparing the two pipelines at those same fitting points does not establish independence. Metric references on the plane, independent acquisition groups and an explicit extrinsic calibration allow the bias to be assessed.

4.3 Line mask, laser candidates and local error

The real detector consumes a BGR image, constructs HSV masks and selects a connected blue-line component. It uses morphological clean-up, connectivity, an elongation preference and area gates. The red laser mask spans both ends of the hue circle. A candidate can be located via contour area and spatial moments $(m_{10},m_{01})/m_{00}$. The retained code also uses circularity $4\pi\ A/P^2$, aspect, area and previous laser position in the candidate score. A green goal marker can select the positive orientation of the local tangent.

The detector extracts line KLT points. An SVD of centred local neighbours estimates the tangent. Orienting it with the goal or previous tangent and smoothing the orientation stabilises a signed offset from the laser to the local line. This is an image-space line–spot relation. A valid calibrated red centroid supplies a second measurement, the absolute metric wall spot. When the tangent of a complex curve changes or the nearest points come from separate branches, the local PCA line may approximate the wrong segment. The ordered curve representation and a spatial or phase gate become important. Projective scale also prevents directly treating a raw pixel offset as a fixed metric error.

Box 4.1 Projective camera model, homography estimation and its uncertainty

Theoretical complement to Sections 4.1–4.2. First-order and Monte Carlo values are computed by theory_ur7e.py on the geometry of Figure 4.1.

Camera model

A world point X maps to the homogeneous pixel $\tilde{u}$ through the projection matrix $P = K[R\,|\,t]$,

$$\lambda\tilde{u} = K[R\,|\,t]\,\tilde{X}, \quad K = \begin{bmatrix} f_x & \gamma & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{B4.1}$$

with skew $\gamma \simeq 0$ for modern sensors. Real lenses add radial and tangential distortion of the normalised coordinates (x, y), $r^2 = x^2 + y^2$ (Brown–Conrady):

$$\begin{aligned} x_d &= x\,\kappa(r) + 2p_1xy + p_2(r^2 + 2x^2), \\ y_d &= y\,\kappa(r) + p_1(r^2 + 2y^2) + 2p_2xy, \\ \kappa(r) &= 1 + k_1r^2 + k_2r^4 + k_3r^6. \end{aligned} \tag{B4.2}$$

The map has no closed-form inverse; undistortion is a fixed-point iteration. Zhang's method [20] estimates K, (k, p) and one pose per view of a planar target by closed-form initialisation followed by nonlinear least squares on the reprojection error.

Plane-induced homography

For wall points $X = p_0 + b_y y + b_z z$, the third column of $[R\,|\,t]$ multiplies zero and (B4.1) reduces to (4.1),

$$H_{W\to I} = K\,[\,Rb_y \;\; Rb_z \;\; Rp_0 + t\,], \tag{B4.3}$$

a rank-3 matrix whenever the camera centre is off the plane. H is defined up to scale and has 8 degrees of freedom. It encodes K and the plane pose jointly; with known K the decomposition $K^{-1}H \propto [r_1\ r_2\ t]$ recovers the pose, with $r_3 = r_1 \times r_2$ after orthonormalisation.

Direct linear transformation

A correspondence $\tilde{u}_i \leftrightarrow \tilde{s}_i$ satisfies $\tilde{u}_i \times H\tilde{s}_i = 0$. Two of its three rows are independent:

$$\begin{bmatrix} -\tilde{s}_i^{\mathsf{T}} & 0^{\mathsf{T}} & u_i\tilde{s}_i^{\mathsf{T}} \\ 0^{\mathsf{T}} & -\tilde{s}_i^{\mathsf{T}} & v_i\tilde{s}_i^{\mathsf{T}} \end{bmatrix} h = 0, \qquad h = \mathrm{vec}(H^{\mathsf{T}}). \tag{B4.4}$$

Stacking $N \geq 4$ points gives $Ah = 0$, solved by $\min \|Ah\|$ subject to $\|h\| = 1$, that is, the right singular vector of the smallest singular value. Hartley's normalisation $\hat{u} = T_u u$ moves the centroid to the origin and scales the mean distance to $\sqrt{2}$; then $H = T_s^{-1}\hat{H}T_u$. It equalises the magnitudes of the entries of A and prevents the large dynamic range of pixel products from dominating the solution. The release code implements exactly this estimator (Section 4.2).

Algebraic and geometric error

The DLT minimises an algebraic residual. The maximum-likelihood estimate under isotropic pixel noise instead minimises the reprojection error

$$\hat{H} = \arg\min_H \sum_i \left\| u_i - \pi(H\tilde{s}_i) \right\|^2, \tag{B4.5}$$

with π the dehomogenisation, typically by Levenberg–Marquardt from the DLT start. The Sampson distance $\epsilon^{\mathsf{T}}(\mathcal{J}\mathcal{J}^{\mathsf{T}})^{-1}\epsilon$, where ϵ is the algebraic residual and $\mathcal{J}$ its Jacobian with respect to the measured coordinates, is a first-order approximation of the geometric error at algebraic cost [19].

First-order uncertainty propagation

Let $x \in \mathbb{R}^{2N}$ stack the measured pixels with $\mathrm{Cov}(x) = \sigma^2 I$. Through the estimator, the wall coordinates $\hat{s}_j = g_j(x)$ of the test points have, to first order,

$$\mathrm{Cov}(\hat{s}) \approx \sigma^2\, GG^{\mathsf{T}}, \qquad G = \left.\frac{\partial g}{\partial x}\right|_{x_0}, \tag{B4.6}$$

and the expected mean-square mapping error over M test points is

$$\mathbb{E}[\mathrm{RMS}^2] \approx \frac{\sigma^2}{M}\ \mathrm{tr}(GG^{\mathsf{T}}). \tag{B4.7}$$

G is obtained here by central differences through the complete normalised DLT. For the oblique view of Figure 4.1 ($\sigma = 0.65$ px, $M = 315$):

Design	(B4.7)	MC r.m.s.	MC median
4 corners	4.57 mm	4.54 mm	4.13 mm
25 fiducials	1.98 mm	1.99 mm	1.88 mm

The Monte Carlo columns use 4000 independent draws. The medians agree with Section 4.2 (4.14 and 1.90 mm), which used a different random stream. The linear model predicts the simulated

errors within 1%, so at this noise level the calibration design can be optimised analytically: for example, by choosing fiducial positions that minimise $\mathrm{tr}(GG^{\mathsf{T}})$ (an A-optimal design) instead of running a Monte Carlo study for each candidate. The ratio of the two designs, 2.31, is close to the $\sqrt{25/4} = 2.5$ expected from averaging alone. The remaining difference reflects the better spatial spread of the four corners relative to the test grid.

Local metric scale

Differentiating $s = \pi(H^{-1}\tilde{u})$ gives the 2×2 pixel-to-wall Jacobian of Section 4.2, with entries $(h_{ij} - s_i h_{3j})/(h_3^{\mathsf{T}}\tilde{u})$. Its singular values are the local millimetre-per-pixel scales, equal to $Z/f = 5.75\,\mathrm{mm/px}$ in the fronto-parallel case and anisotropic under oblique viewing.

Common-mode errors

Suppose the reference spots used to fit H are predicted by FK with a bias, $s_i^{\mathrm{ref}} = s_i + b$. The fitted map absorbs the bias, $\hat{H} \approx T_b H$, with T_b the translation by b. Any later residual $\hat{s}^{\mathrm{cam}} - \hat{s}^{\mathrm{FK}}$ is then centred on zero although both estimates are biased by b. Formally, the difference test has zero power against common-mode errors. Independence requires references outside the FK chain (surveyed fiducials or a tracker), as argued in Section 4.3. The extrinsic hand–eye relation $AX = XB$ between robot and camera motions has the same structure: it identifies X only up to the errors shared by A and B.

Two-ray consistency and joint calibration

The spot is observed twice: as the end of the laser ray (3.1) and as the intersection of the camera ray with the wall. With camera centre C and ray direction $r(u) = R^{\mathsf{T}}K^{-1}\tilde{u}$,

$$s^{\mathrm{cam}}(u) = C + \frac{c - n^{\mathsf{T}}C}{n^{\mathsf{T}}r(u)}\, r(u), \tag{B4.8}$$

which has the same form as p_W in (3.1). Both predictions must coincide at every sample. This yields a joint estimation problem over the beam parameters θ_b (laser origin and direction in F), the plane (n, c) and the camera (K, R, C):

$$\begin{aligned}\min_{\theta}\ & \sum_k \left\| s^{\mathrm{cam}}(u_k;\theta) - p_W(q_k;\theta) \right\|^2_{\Sigma_k^{-1}} \\ & + \sum_j \left\| s^{\mathrm{cam}}(u_j;\theta) - s_j^{\mathrm{ref}} \right\|^2_{\Sigma_j^{-1}}.\end{aligned} \tag{B4.9}$$

The second sum anchors the solution to surveyed fiducials. Without it, (B4.9) has the gauge freedom of the common-mode bias described above. The Hessian of (B4.9) at the optimum gives the parameter covariance, as in (B2.7), and its smallest eigenvalues identify the directions that the chosen robot poses fail to excite.

Outliers and degeneracies

With an outlier fraction ε among candidate correspondences, RANSAC on minimal samples of $m = 4$ points needs

$$N = \left\lceil \frac{\log(1-p)}{\log(1-(1-\varepsilon)^m)} \right\rceil \tag{B4.10}$$

draws to obtain an outlier-free sample with probability p. For $\varepsilon = 0.3$ and $p = 0.99$ this gives $N = 17$; for $\varepsilon = 0.5$, $N = 72$. The consensus set is then refitted by (B4.5). Degenerate configurations make A rank-deficient: three collinear points among four, or all fiducials near a line. They are detected by the ratio of the two smallest singular values of the normalised A, which should stay far from 1.

Where the projective model ends

The homography assumes a planar wall and a pinhole camera after undistortion. Wall curvature with sagitta h over the calibrated area produces an unmodelled out-of-plane displacement. Under an oblique view this maps to an in-plane error of order $h \tan\beta$, where β is the viewing angle.
Rolling-shutter read-out adds a row-dependent time offset $\delta t = \rho\, v$ (with v the image row and ρ the line period), which turns motion into a geometric error of $\|\dot{s}\|\,\delta t$. This is the same mechanism as the timing term of Box 12.1.

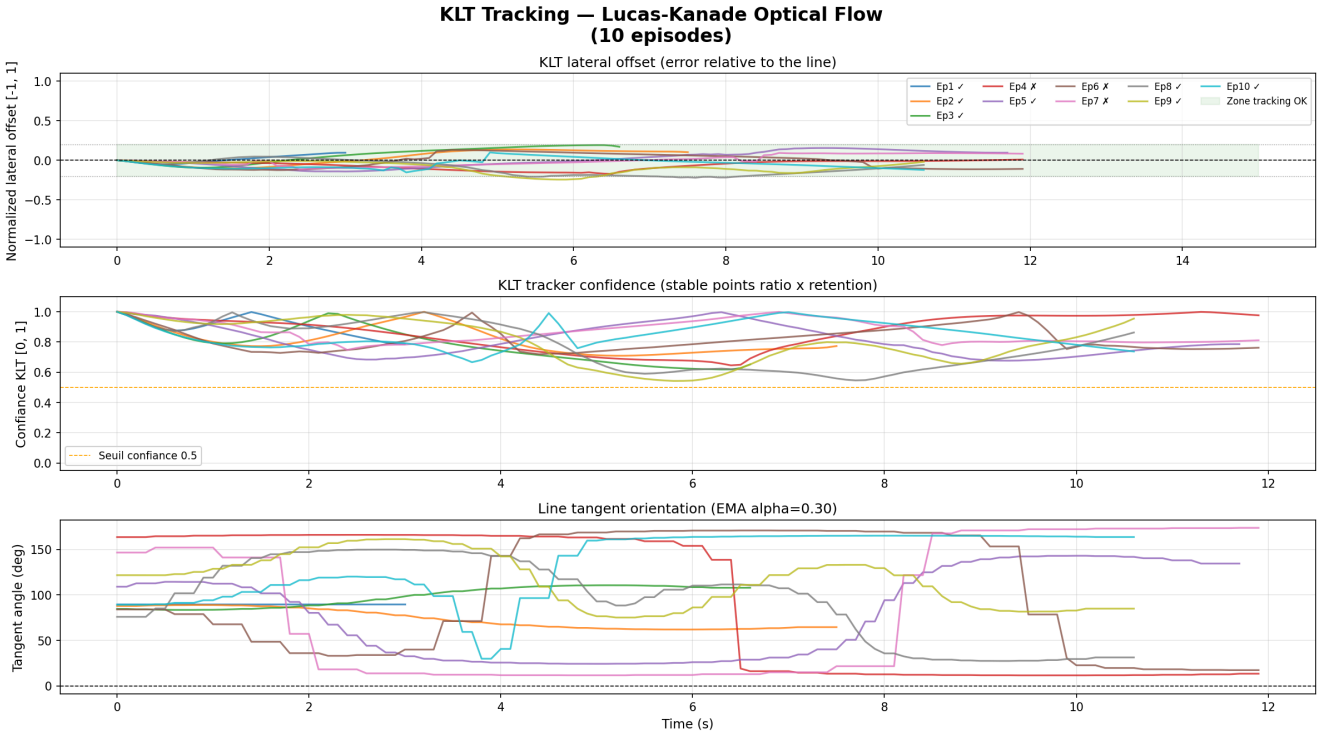


Figure 4.2 – Original KLT tracking diagnostics from the complete-model release [6]. These traces are from the offline simulation program and accompany its simplified observation model.

4.4 The Kanade–Lucas–Tomasi step

For a template patch with brightness $I_0(x)$ and next frame $I_1(x)$, a local shift d is estimated by minimising $\sum_x w(x)[I_1(x+d)-I_0(x)]^2$. At the current shift, linearise about the next-frame gradient g_x. The normal equation for the update is

$$G\,\delta d = -\sum_x w_x g_x r_x,$$
$$G = \sum_x w_x g_x g_x^T, \qquad (4.2)$$
$$r_x = I_1(x+d) - I_0(x).$$

A small eigenvalue of G means a gradient direction carries little information. A straight edge is a limiting case where motion along the edge is poorly observable. Shi–Tomasi scoring selects points with large minimum eigenvalue [21–23]. Pyramids enlarge the capture range by estimating displacement on successively finer levels. Clipping, glare, blurred spots, out-of-image motion and a change of brightness between frames violate the local model or make G poorly conditioned. A forward–backward discrepancy helps detect inconsistent flow, but geometric truth and false association need independent checks.

In the checked real detector, KLT follows features on the blue line and the red spot is selected by HSV contour logic. The notes describe the moving laser marker being lost in the KLT-equipped pipeline. This observation establishes missing or unreliable laser measurements within the overall chain. It leaves the exact cause and the component responsible for the loss unresolved. The real and simulated tunings differ. For example, the presentation records six to 80 simulated points, 25-frame reinitialisation and high confidence near 30 points. The current real detector uses a survival and density score scaled to 45, along with its own masks and update logic [3, 6]. Its confidence is a feature score and has no probability calibration.

4.5 Controlled stress test of spot visibility

The independent image experiment renders a Gaussian red spot of width parameter 2 px, a blue background line, weak texture and Gaussian image noise. A second frame displaces the spot by about 8 px horizontally and up to 4 px vertically. Exposure integration smears it along a horizontal trail of 0 to 10 px, while second-frame gain is 1 or 0.70. A

pyramidal LK test receives the known first-frame centroid and rejects forward–backward errors above 1 px. An HSV centroid test has a known region of interest. Both are scored against the true spot within 1.5 px over 100 trials per combination.

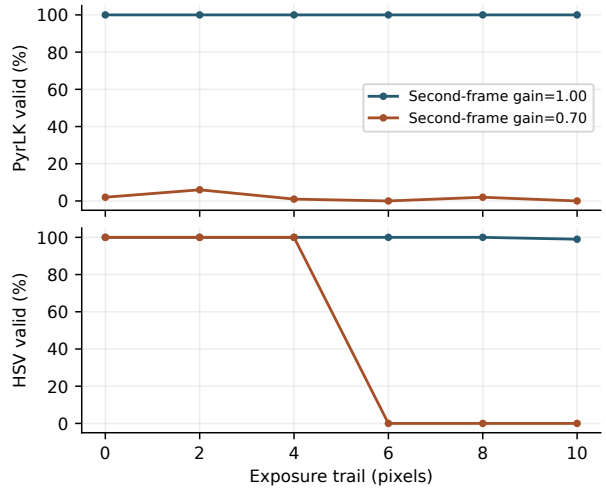


Figure 4.3 – Synthetic photometric and motion-blur stress test. Percentages are fractions of 100 synthetic two-frame trials per blur and gain combination. Both methods receive favourable localisation information.

At unchanged gain, both tests usually find the spot even at a 10 px trail in this model. At gain 0.70 LK retains 0–6% across the levels, and HSV remains at 100% through a 4 px trail before the preset intensity cutoff removes all candidates at 6 px and beyond. This makes two mechanisms separately visible: LK's brightness-constancy sensitivity and the value threshold's interaction with exposure and spatial blur. Actual false positives from red reflections, confidence scores and motion speeds cannot be inferred from these synthetic images. Nor does this experiment replay the project's line-feature KLT node. It is a controlled test of a moving spot with known truth.

5. Fusion and observer consistency

5.1 Two measurement sources on one plane

Use $x=(y,z,v_y,v_z)^T$, with position measured by geometric intersection from encoders and by a calibrated camera. An ideal constant-velocity prediction with timestep Δt and process covariance Q is

$$F=\begin{bmatrix} I_2 & \Delta t I_2 \\ 0 & I_2 \end{bmatrix}, \quad H=[I_2\ \ 0], \quad x_k^- = F x_{k-1}^+, \quad P_k^- = F P_{k-1}^+ F^T + Q. \tag{5.1}$$

The complete-model class is named *LaserDotEKF*. Its internal position observation is linear. Its external spot-velocity input is computed from the nonlinear wall kinematics, and its prediction replaces the velocity entries if the kinematic input is finite. It still propagates covariance using F. The implementation is therefore a linear KF update with a kinematic input and a covariance approximation. A consistent linearisation when resetting velocity from encoders would account for the derivatives of the input and any correlations with the FK measurement.

The default simulation parameters at the component level are Δt=0.004 s, $Q=\mathrm{diag}(4\cdot10^{-6},4\cdot10^{-6},10^{-2},10^{-2})$, $R_{FK}=(0.006)^2$ I, $R_{cam}=(0.018)^2$ I and reset $P_0=\mathrm{diag}(10^{-4},10^{-4},10^{-2},10^{-2})$ [3, 6]. Position-variance units are m²; velocity variances use m²/s². The component defaults and 250 Hz physics clock differ from the nominal SAC action hold of 25 ticks, or 0.10 s plus settle and processing overhead. The offline simulator uses 0.10 s steps. Treating a covariance per step as a covariance per second would change the effective noise model.

For continuous white acceleration of spectral density q_a, integrating with variable Δt gives one model for maintaining consistent units across update intervals:

Box 4.2 Feature tracking, localisation limits and image-based servoing

Theoretical complement to Sections 4.3–4.5. Bounds and noise-free renderings computed by theory_ur7e.py with the image model of Section 4.5.

Optical-flow constraint and the aperture problem

Brightness constancy, $I(x + d, t + 1) = I(x, t)$, linearised to first order gives one scalar equation per pixel,

$$\nabla I(x)^{\mathsf{T}} d + I_t(x) = 0, \tag{B4.11}$$

for the two unknowns of d. Only the component of d along the gradient is observable: this is the aperture problem. Lucas–Kanade pools the constraint over a window with weights w and solves the normal equations (4.2), $G\,d = -\sum_x w\,\nabla I\,I_t$, with the structure tensor

$$G = \sum_x w(x) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}, \quad \lambda_1 \geq \lambda_2 \geq 0. \tag{B4.12}$$

If the residual is white with variance σ_r^2, the estimate has $\mathrm{Cov}(\hat{d}) \approx \sigma_r^2 G^{-1}$. Its worst-direction standard deviation is $\sigma_r / \sqrt{\lambda_2}$, which justifies the Shi–Tomasi criterion $\lambda_2 > \tau$ [22]: corners have two large eigenvalues, an edge has $\lambda_2 \approx 0$, and a flat patch has both small. Iterating the linearisation is a Gauss–Newton method on $\sum_x w\,[I_1(x + d) - I_0(x)]^2$. Each pyramid level halves the displacement, so L levels extend the capture range by about 2^L [23].

Photometric changes

A gain or exposure change violates (B4.11). The affine photometric model $I_1(x + d) = g\,I_0(x) + b$ adds two unknowns,

$$\min_{d,g,b} \sum_x w\,[I_1(x + d) - g\,I_0(x) - b]^2, \tag{B4.13}$$

solved by the same Gauss–Newton scheme with a 4×4 system. Without (g, b) the residual contains $(1 - g)I_0(x)$, which is largest exactly on the bright spot. It is absorbed as a spurious displacement and then rejected by the forward–backward test. This is consistent with the 0–6% LK success at second-frame gain 0.70 in Figure 4.3.

Colour segmentation and moments

The detector of Section 4.3 thresholds $V = \max(R, G, B)$, $S = 1 - \min(R, G, B)/V$ and the hue, an angle that is wrapped at red, so two hue intervals are needed. On the binary mask Ω, the raw moments $m_{pq} = \sum_\Omega x^p y^q$ give the centroid $(m_{10}, m_{01})/m_{00}$. The central moments μ_{pq} give the blob orientation and elongation:

$$\theta = \tfrac{1}{2}\,\mathrm{atan2}(2\mu_{11}, \mu_{20} - \mu_{02}), \quad e = \frac{\lambda_{\max}(\Sigma_\Omega)}{\lambda_{\min}(\Sigma_\Omega)}, \tag{B4.14}$$

with $\Sigma_\Omega = \mu_{..}/m_{00}$ the second-moment matrix. Circularity $4\pi A/P^2$ equals 1 for a disc. A motion trail of length L along x adds $L^2/12$ to the variance along x, so elongation measures blur directly.

Cramér–Rao bound for spot localisation

Model the spot as $\mu(x) = A\exp(-\|x - x_0\|^2/2w^2) + B$ with iid Gaussian pixel noise σ_n. The Fisher information for x_0 is $\mathcal{I} = \sigma_n^{-2} \sum_x \nabla_{x_0}\mu\,\nabla_{x_0}\mu^{\mathsf{T}}$. With dense sampling,

$$\mathcal{I} \approx \frac{A^2}{\sigma_n^2 w^4} \iint (x - x_0)^2 e^{-\|x - x_0\|^2/w^2} = \frac{\pi A^2}{2\sigma_n^2}\, I_2, \tag{B4.15}$$

independent of the width w. Any unbiased estimator therefore satisfies

$$\mathrm{std}(\hat{x}_0) \geq \sigma_n \sqrt{2/\pi}\,/A. \tag{B4.16}$$

For the synthetic images of Section 4.5 ($A = 210$ on the red channel, $\sigma_n = 1.8$) the bound is 0.0068 px, and 0.015 px on the grey image used by LK. Both are two orders of magnitude below the 1.5 px acceptance tolerance. The failures in Figure 4.3 are therefore not limited by noise but by the detector's decision rules.

Why HSV detection collapses at 6 px

A horizontal exposure trail of length L averages the spot along x, so the peak intensity becomes

$$m(L) = \frac{\sqrt{2\pi}\,w}{L}\,\mathrm{erf}\!\left(\frac{L}{2\sqrt{2}\,w}\right) \quad (m(0) = 1). \tag{B4.17}$$

The pixel passes the value test when $g\,(44 + 210\,m) > 135$, that is, $m > (135/g - 44)/210$: 0.433 at gain 1 and 0.709 at gain 0.70. Rendering the image of Section 4.5

without noise gives the peak values and mask sizes below (red-channel model):

L (px)	0	2	4	6	8	10
peak m	0.96	0.92	0.81	0.68	0.55	0.45
mask, $g = 1$	22	22	22	20	16	8
mask, $g = 0.7$	10	8	4	0	0	0

At gain 0.70 the peak crosses the threshold between 4 and 6 px, and the detector needs at least 3 mask pixels. This deterministic cliff reproduces the 100%-to-0% transition of Figure 4.3 without any noise. Detection reliability is therefore governed by a deterministic margin, $g(44 + 210\,m(L)) - 135$. This margin, rather than a signal-to-noise ratio, should be logged and controlled (exposure time, gain or adaptive thresholds) on the real camera.

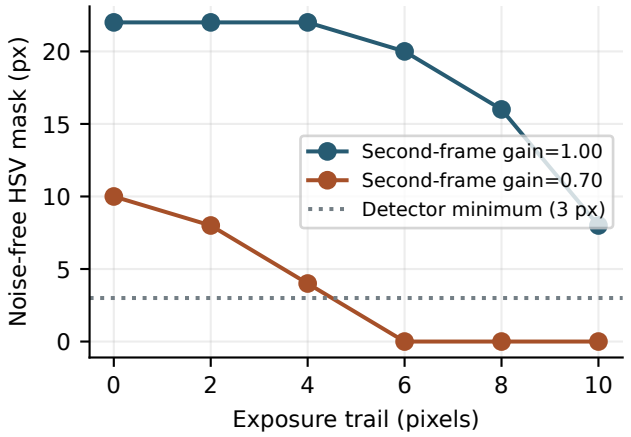


Figure B4.1 – Noise-free HSV mask size versus exposure trail for the image model of Section 4.5. The dotted line is the 3-pixel minimum of the detector.

Image-based visual servoing

For a normalised image point (x, y) at depth Z, the point's image velocity under a camera twist $v_c = (v, \omega)$ is $\dot{s} = L_s v_c$ with the interaction matrix

$$L_s = \begin{bmatrix} -\frac{1}{Z} & 0 & \frac{x}{Z} & xy & -(1+x^2) & y \\ 0 & -\frac{1}{Z} & \frac{y}{Z} & 1+y^2 & -xy & -x \end{bmatrix}. \tag{B4.18}$$

The law $v_c = -k\,\widehat{L_s}^{+} e$ gives $\dot{e} = -k\,L_s\widehat{L_s}^{+} e$. It is locally asymptotically stable if $L_s\widehat{L_s}^{+} \succ 0$, which tolerates substantial errors in the estimated depth $\hat{Z}$.

In the project the camera is fixed (eye-to-hand) and the controlled feature is the laser spot. Its image coordinates obey

$$\dot{u} = J_{\text{img}}(s)\,J_W(q)\,\dot{q}, \tag{B4.19}$$

where J_{img} is the inverse of the pixel-to-wall Jacobian of Box 4.1 and J_W is the wall Jacobian (3.2). The law $\dot{q} = -k\,(\widehat{J_{\text{img}}J_W})^{\dagger}_{\lambda} e_u$ is then a DLS servo in image space. Its convergence condition, $J_{\text{img}}J_W(\widehat{J_{\text{img}}J_W})^{\dagger}_{\lambda} \succ 0$, is the matrix analogue of the small-gain margin (B3.11). Calibration errors in H, the beam or the plane are tolerated as long as this product stays positive definite. The same three error sources that dominate the error budget of Box 12.1 thus also bound the robustness of a camera-closed loop.

Clutter and missed detections

Red reflections create several candidates per frame, and blur or occlusion remove the true one. Probabilistic data association weights the m_k gated candidates instead of choosing one [24]. With detection probability P_D, gate probability P_G, clutter density λ_c and innovations ν_i,

$$\beta_i = \frac{\ell_i}{1 - P_D P_G + \sum_j \ell_j}, \qquad \ell_i = \frac{P_D}{\lambda_c}\,\mathcal{N}(\nu_i; 0, S), \tag{B4.20}$$

and $\beta_0 = 1 - \sum_i \beta_i$ is the probability that no candidate is correct. The update uses the combined innovation $\bar{\nu} = \sum_i \beta_i \nu_i$:

$$\begin{aligned} x^{+} &= x^{-} + K\bar{\nu}, \\ P^{+} &= P^{-} - (1-\beta_0) K S K^{\mathsf{T}} + K\Big(\sum_i \beta_i \nu_i \nu_i^{\mathsf{T}} - \bar{\nu}\bar{\nu}^{\mathsf{T}}\Big) K^{\mathsf{T}}. \end{aligned} \tag{B4.21}$$

The last term inflates the covariance when candidates disagree, the behaviour Section 12.1 asks of the runner instead of treating stale or ambiguous spots as fresh. Logged values of m_k, P_D versus speed and the margin of Figure B4.1 are the quantities needed to fit P_D and λ_c for the real camera.

$$Q(\Delta t) = q_a \begin{bmatrix} \Delta t^3 I_2/3 & \Delta t^2 I_2/2 \\ \Delta t^2 I_2/2 & \Delta t I_2 \end{bmatrix}. \tag{5.2}$$

The numerical values above are tuning settings. They are neither a measurement of the true covariance nor transferable across the simulator and hardware.

5.2 Kalman update and camera acceptance

For a measurement z_k with covariance R, the innovation is $\nu = z_k - H\hat{x}^-$, with covariance S. Minimise posterior trace over linear gains to obtain $K = P^- H^T S^{-1}$. Substituting this gain into the estimator gives the Joseph form:

$$\begin{aligned} S &= HP^- H^T + R, \\ x^+ &= x^- + K\nu, \quad A = I - KH, \\ P^+ &= AP^- A^T + KRK^T. \end{aligned} \tag{5.3}$$

The $\nu^T S^{-1} \nu$ statistic (NIS) has a χ^2 law with two degrees of freedom for the linear Gaussian model and a consistent predictive covariance [18, 24]. A 99% illustrative threshold is 9.21. Gating can reject isolated measurements far from the prediction. Biased calibration, correlated sources and stale timestamps can inflate NIS persistently. Use per-source innovations, accepted count, out-of-sequence rate and age. The matching position-only NEES in the simulation compares $(x_{true} - \hat{x})_{1:2}$ and its 2-D marginal covariance, so its calibration requires independent truth. Computing it from FK used by the filter would overstate consistency.

If two independent simultaneous measurements observe the same point, assimilating them sequentially is equivalent to stacking them. In a static fusion without a prior, the two precisions add. The 6 mm FK and 18 mm camera defaults give 0.9 and 0.1 of the static weight and a combined standard deviation of 5.69 mm. A fixed 10 mm FK bias would still contribute 9 mm. Asynchronous measurements require time alignment or propagation to their own acquisition timestamps. Correlated calibration errors require the cross-covariance in the stacked R or another conservative fusion method. Line-relative KLT information is distinct from the absolute spot measurement; the current code explicitly refuses the former as an absolute camera update.

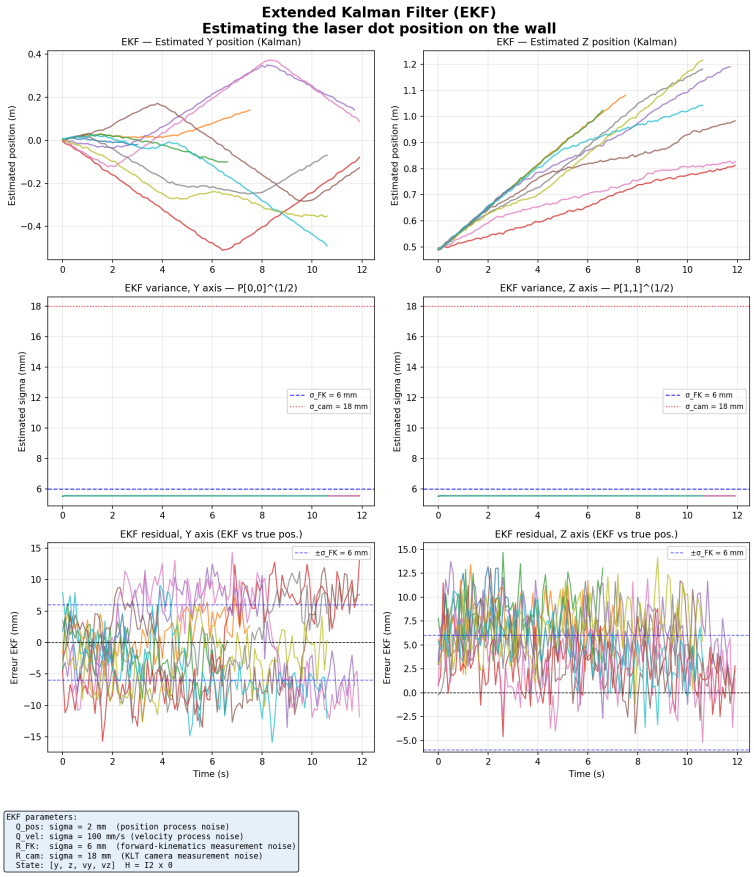


Figure 5.1 – Original simulation filter estimates and uncertainty diagnostics from the complete-model release [6]. Simulated observations and source-defined covariance supply the plotted records.

5.3 Numerical test with missing and contaminated images

The new observer experiment declares a 5 s sinusoidal truth trace, FK observations at 100 Hz with 6 mm iid noise, and camera observations at 25 Hz with 18 mm iid noise. A smooth FK bias approaches (12,−9) mm around 2 s. The camera drops out from 1.8 to 2.6 s. An 8% probability of an additive (90,−75) mm gross error applies on active frames. The same draws drive three comparisons: FK-only, both sensors without rejection and both sensors with a 2-D NIS gate. The synthetic filter uses a discretised acceleration covariance and Joseph update. The experiment omits laser geometry errors so each source of corruption is explicit.

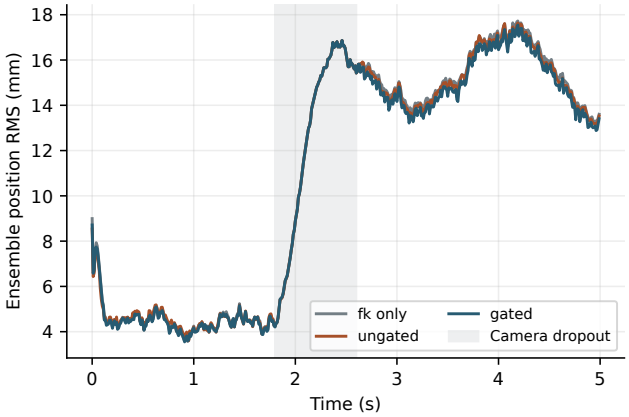


Figure 5.2 – Synthetic observer comparison over 120 repetitions with known truth. The shaded interval indicates missing camera measurements. Position RMS is evaluated across trials at each time.

The overall position RMS values are 12.25, 12.18 and 12.05 mm in that order. The gate rejects 970 of 974 injected bad camera observations and rejects another 200. The small RMS benefit is consistent with the dominant FK bias and lower precision and rate of the camera observations. A robust update helps reject outliers while a measurement bias

Box 5.1 Kalman filtering: derivation, steady state and consistency

Theoretical complement to Sections 5.1–5.2. Steady-state values computed by theory_ur7e.py for the default parameters of Section 5.1.

Model and Bayesian recursion

The observer of Section 5 is the linear Gaussian state-space model

$$\begin{aligned} x_{k+1} &= F x_k + w_k, \qquad & z_k &= H x_k + v_k, \\ w_k &\sim \mathcal{N}(0, Q), & v_k &\sim \mathcal{N}(0, R). \end{aligned} \tag{B5.1}$$

The posterior $p(x_k \mid z_{1:k})$ obeys the Chapman–Kolmogorov prediction and Bayes' update,

$$\begin{aligned} p(x_k \mid z_{1:k}) \propto p(z_k \mid x_k) \int p(x_k \mid x_{k-1}) \\ \times\, p(x_{k-1} \mid z_{1:k-1})\, dx_{k-1}, \end{aligned} \tag{B5.2}$$

and remains Gaussian, $\mathcal{N}(\hat{x}_k, P_k)$, so the recursion reduces to two moments [24].

The update as an optimisation

The posterior mean is the minimiser of

$$\mathcal{J}(x) = \|x - x^-\|^2_{(P^-)^{-1}} + \|z - Hx\|^2_{R^{-1}}. \tag{B5.3}$$

Setting $\nabla \mathcal{J} = 0$ gives the information form $(P^+)^{-1} = (P^-)^{-1} + H^\mathsf{T} R^{-1} H$ and, by the matrix inversion lemma, the gain form of (5.3),

$$K = P^- H^\mathsf{T} (H P^- H^\mathsf{T} + R)^{-1}, \qquad x^+ = x^- + K\nu. \tag{B5.4}$$

Without the Gaussian assumption the same K minimises $\operatorname{tr} P^+$ among linear unbiased updates, so the filter is the best linear unbiased estimator for any noise with these covariances. For an arbitrary gain, $P^+ = (I - KH)P^-(I - KH)^\mathsf{T} + KRK^\mathsf{T}$ (Joseph form) is exact and stays symmetric positive semidefinite in floating point; the simplified $P^+ = (I - KH)P^-$ is valid only at the optimal K. Because the information form is additive, sequential updates with independent sensors equal the stacked update, as stated in Section 5.2.

Riccati equation and steady state

Eliminating P^+ gives the discrete Riccati recursion for the prior covariance,

$$\begin{aligned} P_{k+1} = F P_k F^\mathsf{T} + Q \\ - F P_k H^\mathsf{T} (H P_k H^\mathsf{T} + R)^{-1} H P_k F^\mathsf{T}. \end{aligned} \tag{B5.5}$$

If (F, H) is detectable and $(F, Q^{1/2})$ stabilisable, P_k converges to the unique stabilising solution of the algebraic Riccati equation, and the error dynamics $(I - KH)F$ are then asymptotically stable. For the constant-velocity model of (5.1), one axis has

$$\mathcal{O} = \begin{bmatrix} H \\ HF \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & \Delta t \end{bmatrix}, \qquad \operatorname{rank} \mathcal{O} = 2, \tag{B5.6}$$

so position measurements alone observe the velocity. With the defaults $\Delta t = 4$ ms, $Q = \operatorname{diag}(4 \cdot 10^{-6}\,\mathrm{m}^2, 10^{-2}\,\mathrm{m}^2/\mathrm{s}^2)$ and $R = (6\,\mathrm{mm})^2$, the steady state is:

	FK only	FK + camera
Equivalent $\sqrt{R}$ (mm)	6.00	5.69
Prior position std (mm)	4.79	4.66
Posterior position std (mm)	3.74	3.60
Gains (α, β)	(0.390, 0.052)	(0.401, 0.054)
$\lvert\mathrm{eig}((I - KH)F)\rvert$	0.781	0.774

The steady-state filter is exactly an α–β tracker, $\hat{p}^+ = \hat{p}^- + \alpha\nu$ and $\hat{v}^+ = \hat{v}^- + (\beta/\Delta t)\nu$. Its error pole 0.781 per 4 ms period corresponds to a time constant of about 16 ms. Adding the camera at every step changes the posterior standard deviation by only 4%, the precision ratio of 9:1 stated in Section 5.2.

Process noise from continuous time

For a continuous white-noise acceleration $\ddot{p} = a(t)$ with $\mathbb{E}[a(t)a(\tau)] = q_a \delta(t - \tau)$, discretisation over Δt gives

$$\begin{aligned} Q(\Delta t) &= \int_0^{\Delta t} e^{A\tau} G G^\mathsf{T} e^{A^\mathsf{T}\tau} q_a \, d\tau \\ &= q_a \begin{bmatrix} \Delta t^3/3 & \Delta t^2/2 \\ \Delta t^2/2 & \Delta t \end{bmatrix}, \end{aligned} \tag{B5.7}$$

with $A = \left[\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right]$ and $G = [0\ 1]^\mathsf{T}$, which is (5.2). The entries scale as Δt^3, Δt^2 and Δt, so the model is invariant to the sampling period. The diagonal per-step default Q has no such scaling: halving Δt doubles the process noise injected per second. This is the formal content of the warning in Section 5.1.

Consistency tests

For a consistent filter, the normalised estimation

error and the normalised innovation satisfy

$$\epsilon_k = \tilde{x}_k^{\mathsf{T}} P_k^{-1} \tilde{x}_k \sim \chi_n^2, \qquad \eta_k = \nu_k^{\mathsf{T}} S_k^{-1} \nu_k \sim \chi_m^2. \tag{B5.8}$$

Averaged over N independent samples, $N\bar{\eta} \sim \chi_{Nm}^2$, which gives the two-sided acceptance interval

$$\bar{\eta} \in [\chi_{Nm}^2(0.025),\, \chi_{Nm}^2(0.975)]/N \approx m \pm 1.96\sqrt{2m/N}. \tag{B5.9}$$

Whiteness is checked through the normalised autocorrelation of the innovations, which should lie within $\pm 1.96/\sqrt{K}$ for K samples. NEES requires the truth, which only the simulation provides (Section 5.2). NIS is available on hardware and is the operational test.

Identifying Q and R from logged innovations

Section 5.1 stresses that the numerical covariances are tuning settings. They can be estimated from data, because the innovation sequence of a suboptimal filter is coloured in a way that depends on (Q, R). For a stationary filter with gain K, the innovation autocovariances $C_j = \mathbb{E}[\nu_k \nu_{k-j}^{\mathsf{T}}]$ satisfy

$$\begin{aligned} C_0 &= HPH^{\mathsf{T}} + R, \\ C_j &= H[F(I - KH)]^{j-1} F(PH^{\mathsf{T}} - KC_0), \quad j \geq 1, \end{aligned} \tag{B5.10}$$

which are linear in the unknown entries of Q and R through P. Stacking sample estimates $\hat{C}_j$ gives a least-squares problem (the autocovariance least-squares method). A simpler adaptive estimate of the measurement noise is the innovation-based

$$\hat{R} = \frac{1}{N} \sum_k (\nu_k \nu_k^{\mathsf{T}} - H P_k^- H^{\mathsf{T}}), \tag{B5.11}$$

which is unbiased when Q and P^- are correct. On hardware logs, (B5.10)–(B5.11) turn the NIS test of (B5.9) from a pass/fail check into an estimate of what is wrong.

Tuning by a single tracking index

For the discrete white-noise acceleration model, in which the acceleration is constant over each period with standard deviation σ_w,

$$Q = \sigma_w^2 \begin{bmatrix} \Delta t^4/4 & \Delta t^3/2 \\ \Delta t^3/2 & \Delta t^2 \end{bmatrix}, \tag{B5.12}$$

the steady-state gains depend only on the tracking index $\Lambda = \sigma_w \Delta t^2 / \sigma_v$ (Kalata):

$$\begin{aligned} \alpha &= -\tfrac{1}{8}(\Lambda^2 + 8\Lambda - (\Lambda + 4)\sqrt{\Lambda^2 + 8\Lambda}), \\ \beta &= \tfrac{1}{4}(\Lambda^2 + 4\Lambda - \Lambda\sqrt{\Lambda^2 + 8\Lambda}), \end{aligned} \tag{B5.13}$$

with $\Lambda^2 = \beta^2/(1 - \alpha)$. The script reproduces the Riccati gains to machine precision for $\Lambda = 0.05$, 0.5 and 2; for example, $\Lambda = 0.5$ gives $\alpha = 0.628$ and $\beta = 0.305$. Choosing Q then reduces to choosing one physically interpretable number, the expected spot acceleration relative to the sensor noise, instead of four independent diagonal entries.

Smoothing for offline evaluation

Offline analyses of logged runs, such as the hardware tests proposed in Section 12.3, can use all the data. The Rauch–Tung–Striebel smoother runs backwards after the filter,

$$\begin{aligned} C_k &= P_k^+ F^{\mathsf{T}} (P_{k+1}^-)^{-1}, \\ \hat{x}_k^s &= \hat{x}_k^+ + C_k(\hat{x}_{k+1}^s - \hat{x}_{k+1}^-), \\ P_k^s &= P_k^+ + C_k(P_{k+1}^s - P_{k+1}^-)C_k^{\mathsf{T}}, \end{aligned} \tag{B5.14}$$

with $P_k^s \preceq P_k^+$. Smoothed trajectories are the appropriate reference for scoring a recorded path, whereas filtered estimates are the information actually available to a controller in real time.

Box 5.2 Bias, correlation and robust fusion: analysis of the observer experiment

Theoretical complement to Sections 5.2–5.3. The observer test of Section 5.3 is re-run with the same random stream and additional diagnostics in theory_ur7e.py.

Diagnosing the gate with NIS

Re-running the experiment of Section 5.3 reproduces its counts exactly: 1170 camera frames rejected, of which 970 were injected outliers. There are 11 506 valid camera frames, so a consistent filter with a 99% gate should reject about 115 of them. It rejects 200. Splitting the valid frames by time locates the excess:

Valid camera frames	N	mean NIS	rejected
$t < 1.5$ s (no FK bias)	4222	2.008	0.85%
$t > 2.6$ s (bias settled)	6500	2.644	2.37%

Before the bias appears, the mean NIS lies inside the acceptance interval (B5.9), 2 ± 0.060, and the rejection rate is close to 1%. After it settles, the mean NIS of 2.644 lies far outside 2 ± 0.049. The filter is then inconsistent although every model parameter is unchanged.

Non-central χ^2 explanation

If the innovation carries a deterministic offset b_ν, its NIS follows a non-central $\chi^2_2(\lambda)$ distribution with $\lambda = b_\nu^{\mathsf{T}} S^{-1} b_\nu$, and the gate rejects with probability

$$P_{\mathrm{rej}}(\lambda) = \Pr[\chi^2_2(\lambda) > 9.21] = Q_1(\sqrt{\lambda}, \sqrt{9.21}), \tag{B5.15}$$

where Q_1 is Marcum's Q function. The FK-dominated estimate inherits the 15 mm bias $(12, -9)$ mm. With $S \approx (\sigma^2_{\mathrm{cam}} + \sigma^2_{\mathrm{FK}})I$, this gives $\lambda \le 0.625$ and $P_{\mathrm{rej}} \le 2.7\%$, consistent with the observed 2.37%. The excess rejections are therefore the statistical signature of the unmodelled FK bias. A gate tuned for outliers silently discards valid frames once a bias develops, which is exactly when the camera information is most needed.

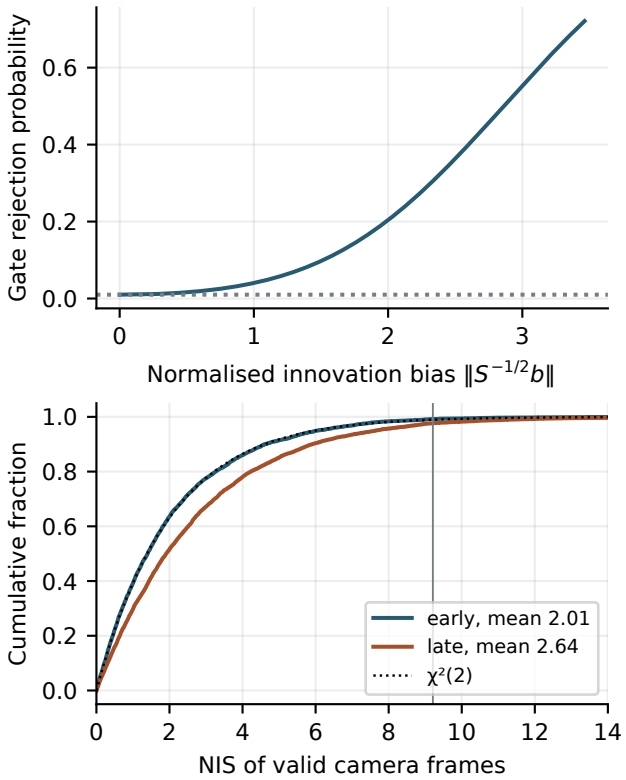


Figure B5.1 – Top: gate rejection probability versus normalised innovation bias, with the 1% nominal level. Bottom: empirical distribution of the NIS of valid camera frames before and after the FK bias, compared with $\chi^2(2)$.

Bias augmentation and observability

Model the bias explicitly per axis, $x = [p,\ v,\ b]^{\mathsf{T}}$ with $b_{k+1} = b_k + w_b$, $z^{\mathrm{FK}} = p + b + v^{\mathrm{FK}}$ and $z^{\mathrm{cam}} = p + v^{\mathrm{cam}}$. The observability matrix $[H; HF; HF^2]$ has rank 2 with FK alone and rank 3 with both sensors. The bias is therefore estimable only when an independent absolute measurement is present. This is the formal version of the requirement in Sections 4.3 and 12.1 that FK registration be checked against references outside the FK chain. When b is not estimated but its uncertainty is known, the Schmidt (consider) filter propagates the cross-covariance P_{xb} without updating b. The result is a consistent but more conservative P.

Measurement gaps and gate width

Without measurements the prior grows as $P(\tau) = F^n P^+ F^{n\mathsf{T}} + \sum_{j<n} F^j Q F^{j\mathsf{T}}$ with $n = \tau/\Delta t$. With the per-step default Q of Section 5.1, the position standard deviation reaches 41.5 mm after a 0.1 s gap (one action period) and 687 mm after 0.8 s. The gate (B5.8) scales with $S = HP(\tau)H^\mathsf{T} + R$. The injected gross error $(90, -75)$ mm, of norm 117 mm, has an NIS of $117^2/(41.5^2 + 18^2) = 6.7 < 9.21$ after a single 0.1 s gap and would be accepted. A NIS gate therefore protects only while the prior is tight. After dropouts, reacquisition should rely on an independent consistency check, such as the FK prediction with its own covariance or a temporal persistence test, before the gate is trusted again.

Unknown correlation: covariance intersection

If two estimates (x_a, P_a) and (x_b, P_b) share an unknown correlation, for example through a common calibration, the stacked update over-counts information. Covariance intersection,

$$\begin{aligned} P_{\mathrm{CI}}^{-1} &= \omega P_a^{-1} + (1-\omega) P_b^{-1}, \\ x_{\mathrm{CI}} &= P_{\mathrm{CI}}(\omega P_a^{-1} x_a + (1-\omega) P_b^{-1} x_b), \end{aligned} \tag{B5.16}$$

with $\omega \in [0, 1]$ chosen to minimise $\operatorname{tr} P_{\mathrm{CI}}$ or $\det P_{\mathrm{CI}}$, is consistent for every cross-correlation. This is one "conservative fusion method" that Section 5.2 mentions.

Nonlinear measurement of the spot

If the filter state is the joint vector, the camera observes $z = s(q) + v$ through the laser geometry. The extended Kalman filter uses $H_k = \partial s/\partial q = J_W(\hat{q}_k^-)$ from (3.2). The neglected second-order term $\frac{1}{2}\delta q^\mathsf{T} \nabla^2 s\, \delta q$ grows with the curvature of the ray–plane map, and hence at grazing incidence through P_d. There the iterated EKF (relinearising at x^+) or the unscented transform with $2n + 1$ sigma points reduces the linearisation bias.

Delay and robust updates

A camera frame that is ℓ periods old can be fused exactly by augmenting the state with $x_{k-\ell}$, or by storing the history and re-filtering from $k - \ell$. The second option is cheaper at low rates. Box 3.2 shows why this matters in closed loop. For heavy-tailed errors, the Huber update replaces $K\nu$ by $K\,\psi(\nu)$ with the weight

$$\psi(\nu) = \nu \, \min\!\left(1, c/\sqrt{\nu^\mathsf{T} S^{-1} \nu}\right), \tag{B5.17}$$

so gross errors are down-weighted rather than discarded. This uses the same normalised loss as the reward of Section 10.2.

remains a calibration or estimation-state problem. Individual NIS thresholds in the real node differ and require actual noise, timing and false-positive data for validation.

6. Reinforcement learning formulations

6.1 State, history and reward

An MDP uses state s, bounded action a, transition kernel p(s'|s,a), reward r and discount γ [25]. A policy $\pi_\theta(a|s)$ selects actions and maximises expected discounted return $G_t=\sum_{j\geq 0}\gamma^j r_{t+j}$. State activation depends on the task: joint coordinates and ordered trajectory progress for the simplified simulator; empirical state, calibration and visual information for a real camera pipeline. Occlusion, exposure, delayed feedback and unobserved velocity make the deployed problem partially observable. The filter, age of observations and observation history can approximate the belief over those latent variables.

A dense reward can be decomposed into target distance, lateral error, progress, saturation and action variation. In the simplified training environment the initial approach receives distance penalties and an acquisition bonus. Tracking adds local nearest-vertex error, the current waypoint distance, a fine-error bonus, cursor bonuses and a change-in-action penalty. If increasing the cursor is rewarded even when final placement is unresolved, return and strict completion differ. An explicit terminal geometric criterion is therefore needed when comparing agents. Reward magnitudes also depend on resets, clipping and action interval.

Potential-based shaping provides an instructive reference. For $F(s,s')=\gamma\Phi(s')-\Phi(s)$, the discounted sum telescopes. With bounded Φ or a consistent terminal potential, $G'_t=G_t-\Phi(s_t)$ and the optimal policy is preserved [26]. General additions such as independent progress bonuses, distance-shaping and altered success gates in the project change the practical optimisation target. They may help training while needing direct evaluation on the stated physical task.

6.2 Soft Actor-Critic

SAC augments the expected return by the entropy of the selected policy [27, 28]. With temperature $\alpha>0$, the objective is the discounted expectation of $r_t-\alpha\log\pi_\theta(a_t|s_t)$. The soft value for a policy is an expected soft Q with the same entropy term:

$$\begin{aligned} V^\pi(s) &= \mathbb{E}_{a\sim\pi} \\ &[Q^\pi(s,a)-\alpha\log\pi(a|s)], \\ Q^\pi(s,a) &= r(s,a)+\gamma\mathbb{E}_{s'}V^\pi(s'). \end{aligned} \tag{6.1}$$

The target for two critic networks $Q_{\varphi i}$ and their slowly updated copies is

$$\begin{aligned} y &= r+\gamma(1-d) \\ &[\min_i Q_{\bar{\phi}_i}(s',a') \\ &-\alpha\log\pi_\theta(a'|s')]. \end{aligned} \tag{6.2}$$

Here $a'\sim\pi_\theta(\cdot|s')$ and d=1 denotes a true absorbing end. A time-limit truncation is separate if the underlying model allows a bootstrap beyond the horizon. Both critics minimise mean squared Bellman residual. The minimum curbs some positive value bias; its quality still depends on data support, reward scale and critic error.

A reparameterised stochastic actor emits a Gaussian latent action $u=\mu_\theta(s)+\sigma_\theta(s)\odot\varepsilon$, then $a=\tanh u$. Change of variables yields

$$\begin{aligned} &\log\pi_\theta(a|s) \\ &=\log p_\theta(u|s) \\ &-\sum_j \log(1-\tanh^2 u_j). \end{aligned} \tag{6.3}$$

The last term is essential when evaluating entropy in bounded action coordinates. For stable numerical implementation the logarithm is evaluated with a suitable transformation or near-limit guard. A replay-batch actor update minimises $\mathbb{E}[\alpha\log\pi_\theta(a|s)-\min_i Q_{\varphi i}(s,a)]$.
Temperature can be fixed or adjusted towards an entropy target, and critic targets are updated by $\bar{\varphi}\leftarrow(1-\tau)\bar{\varphi}+\tau\varphi$. This is off-policy learning because past transitions remain useful for Q regressions. Frozen deterministic evaluation uses $\tanh\mu_\theta(s)$. Training entropy and exploration

statistics cannot be inferred from the deterministic exported actor.

The public simple checkpoint uses two hidden layers of width 512 and six tanh actions [8]. Its saved step counter is 1,900,000. The slide description rounds the training budget to 2 million. The complete observation V5 architecture uses width 256 and two actions, with a 60,000-step offline artifact. The real folder distributes a separate 40,000-step 2-action checkpoint. Different input metadata and scaling prevent assuming that these actors are interchangeable.

6.3 Proximal Policy Optimization

PPO forms an empirical estimate of an advantage $A_t=Q^{\pi}(s_t,a_t)-V^{\pi}(s_t)$ from rollouts of the previous policy [29]. With critic V, $\delta_t=r_t+\gamma(1-d_t)V(s_{t+1})-V(s_t)$. Generalised advantage estimation with λ_{GAE} discounts the temporal residuals:

$$\widehat{A}_t=\sum_{\ell\geq 0}(\gamma\lambda_{\mathrm{GAE}})^{\ell}\delta_{t+\ell}. \tag{6.4}$$

At a true terminal state the recursion ends. Large λ_{GAE} increases rollout sensitivity. A smaller value relies more on the approximate critic. With probability ratio $r_t(\theta)=\pi_\theta(a_t|s_t)/\pi_{old}(a_t|s_t)$, PPO's clipped actor objective is

$$\mathbb{E}\min\{r_t\widehat{A}_t,\ \mathrm{clip}(r_t,1-\varepsilon,1+\varepsilon)\widehat{A}_t\}. \tag{6.5}$$

Clipping caps the surrogate incentive to push ratios beyond the interval in the favourable direction. It is a coordination mechanism for multiple optimisation passes on collected rollouts. It gives no absolute limit on all changes to the network or the realised robot. By comparison, SAC repeatedly reuses off-policy transitions. A comparison at equal environment interactions addresses sample usage with the chosen hyperparameters and reward. Wall-clock cost, asymptotic performance and robustness to reward changes require separate comparisons [30, 31].

6.4 Goal conditioning, HER and demonstrations

A goal-conditioned policy π(a|s,g) conditions on desired goal g. In a sparse setting, use achieved coordinate h(s') and radius ε_g:

$$r(s,a,s';g)=-\mathbf{1}\{\|h(s')-g\|_2>\varepsilon_g\}. \tag{6.6}$$

HER draws a goal $g'=h(s_j)$ from a later achieved state in the same episode and recomputes the reward for stored s,a,s'. Motion data acquired during a failed original task can then update the critic using an achieved goal [32]. This is valid for shared dynamics that do not depend on goal labels, provided the relabelled records use a consistent observation, reward function and terminal handling. Finite histories, resets, goal-conditioned termination and noise in the stored achieved coordinate need explicit treatment. Computing a dense shaping term separately from the relabelled reward would give inconsistent targets.

The UR5e package uses *HerReplayBuffer* with the future strategy and four sampled goals per transition setting. It also injects scripted demonstrations. Demonstration warm start prepopulates replay with actual transitions. It has a different role from relabelling and does not by itself optimise an imitation loss. In the checked complete wall-following trainer, a standard SAC replay buffer receives optional scripted trajectories; the repository overview mentions HER, but the observed training call does not install a goal-conditioned HER buffer. HER is verified explicitly for the reaching variant and the UR5e pick-and-place branch. Training logs and checkpoints are interpreted against these executable implementations.

Curriculum changes the state or target distribution over time. Start with shorter, smoother paths, larger terminal radii or more helpful resets, then broaden difficulty. A curriculum run can show a lower success fraction when its tasks become harder even as the policy improves on fixed tests. A valid comparison stores the curriculum level with each record and fixes held-out paths and evaluation rules independently of the stage scheduler.

Box 6.1 Mathematical foundations of the learning formulations

Theoretical complement to Sections 6–7 and 11.2. Standard results stated in the notation of Section 6, with the project's hyperparameters where they apply.

Bellman operators and contraction

For a policy π, the evaluation operator acts on bounded functions Q as

$$(\mathcal{T}^{\pi}Q)(s,a) = r(s,a) + \gamma\, \mathbb{E}_{s'\sim p(\cdot|s,a),\, a'\sim\pi(\cdot|s')}[Q(s',a')], \tag{B6.1}$$

and the optimality operator $\mathcal{T}^{*}$ replaces the expectation over a' by a maximum. Both are γ-contractions in the supremum norm,

$$\|\mathcal{T}Q_1 - \mathcal{T}Q_2\|_{\infty} \le \gamma\, \|Q_1 - Q_2\|_{\infty}, \tag{B6.2}$$

so by Banach's fixed-point theorem each has a unique fixed point (Q^{π}, respectively Q^{*}), and iteration converges geometrically [25]. Rewards bounded by $r_{\max}$ give $\|Q\|_{\infty} \le r_{\max}/(1-\gamma)$. For $\gamma = 0.99$, the effective horizon $1/(1-\gamma)$ is 100 decisions, or 10 s at the 0.1 s action period of the wall controller.

Maximum-entropy objective

SAC optimises $J(\pi) = \sum_t \mathbb{E}[\gamma^t (r_t + \alpha\, \mathcal{H}(\pi(\cdot|s_t)))]$. The soft Bellman operator is

$$(\mathcal{T}^{\pi}_{\text{soft}}Q)(s,a) = r + \gamma\, \mathbb{E}[Q(s',a') - \alpha \log \pi(a'|s')], \tag{B6.3}$$

again a γ-contraction. Its optimal policy is a Boltzmann distribution over the soft Q function,

$$\begin{aligned} \pi^{*}(a|s) &= \exp\big((Q^{*}(s,a) - V^{*}(s))/\alpha\big), \\ V^{*}(s) &= \alpha \log \int \exp(Q^{*}(s,a)/\alpha)\, da, \end{aligned} \tag{B6.4}$$

which tends to the greedy policy as $\alpha \to 0$. Soft policy iteration alternates evaluation with the projection

$$\pi_{\text{new}} = \arg\min_{\pi'\in\Pi} D_{\mathrm{KL}}\Big(\pi'(\cdot|s) \,\Big\|\, \frac{\exp(Q^{\pi_{\text{old}}}(s,\cdot)/\alpha)}{Z(s)}\Big), \tag{B6.5}$$

which guarantees $Q^{\pi_{\text{new}}} \ge Q^{\pi_{\text{old}}}$ for tabular problems [27]. The three SAC losses are the sample versions of this scheme:

$$\begin{aligned} J_Q(\phi_i) &= \mathbb{E}\big[(Q_{\phi_i}(s,a) - y)^2\big], \\ J_{\pi}(\theta) &= \mathbb{E}\big[\alpha \log \pi_\theta(a|s) - \min_i Q_{\phi_i}(s,a)\big], \\ J(\alpha) &= \mathbb{E}\big[-\alpha(\log \pi_\theta(a|s) + \bar{\mathcal{H}})\big], \end{aligned} \tag{B6.6}$$

with y from (6.2) and the reparameterised action of (6.3), which makes J_π differentiable through $a = \tanh(\mu_\theta + \sigma_\theta \odot \varepsilon)$. The usual entropy target is $\bar{\mathcal{H}} = -\dim \mathcal{A}$: -6 for the simple follower, -2 for the wall actor and -4 for the UR5e branch. Taking the minimum of two critics counters the overestimation bias $\mathbb{E}[\max_a \hat{Q}] \ge \max_a \mathbb{E}[\hat{Q}]$ produced by maximising noisy estimates.

Policy-gradient bounds behind PPO

The performance-difference lemma relates two policies exactly,

$$J(\pi') - J(\pi) = \frac{1}{1-\gamma}\, \mathbb{E}_{s\sim d^{\pi'},\, a\sim\pi'}[A^{\pi}(s,a)], \tag{B6.7}$$

where $d^{\pi'}$ is the discounted state distribution of the new policy. Replacing $d^{\pi'}$ by d^{π} gives the surrogate $L_\pi(\pi')$, and the trust-region bound

$$J(\pi') \ge L_\pi(\pi') - \frac{4\epsilon\gamma}{(1-\gamma)^2}\, \max_s D_{\mathrm{KL}}(\pi(\cdot|s)\,\|\,\pi'(\cdot|s)), \tag{B6.8}$$

with $\epsilon = \max_{s,a} |A^{\pi}|$, guarantees improvement for small steps. PPO's clipping in (6.5) is a first-order surrogate for this KL constraint [29]. GAE (6.4) interpolates between the one-step temporal difference ($\lambda = 0$: low variance, biased by the critic) and the Monte Carlo return ($\lambda = 1$: unbiased, high variance), since

$$\hat{A}_t^{\mathrm{GAE}(\gamma,\lambda)} = \sum_{\ell\ge 0} (\gamma\lambda)^{\ell}\, \delta_{t+\ell} = G_t^{\lambda} - V(s_t), \tag{B6.9}$$

where G_t^{λ} is the λ-return.

Why hindsight relabelling is valid

In the goal-augmented MDP with state (s,g), the transition $p(s'|s,a)$ does not depend on g, and g enters only through $r(s,a,s';g)$. Any stored (s,a,s') can therefore be combined with any goal g' and the recomputed reward $r(s,a,s';g')$ to give a sample from the same MDP [32]. Off-policy critic targets remain correct. What changes is the distribution of goals in the replay buffer: with the "future" strategy and $k = 4$ relabelled goals per transition, about 80% of samples use achieved goals. HER therefore

reshapes the training distribution without biasing the dynamics. This is the reason it must be paired with a matching sparse reward and termination rule (Section 6.4 and Figure 11.2).

Shaping invariance

For $F(s, s') = \gamma\Phi(s') - \Phi(s)$, the shaped return of any trajectory telescopes:

$$\sum_{t\geq 0} \gamma^t (r_t + F_t) = \sum_{t\geq 0} \gamma^t r_t - \Phi(s_0), \tag{B6.10}$$

provided $\gamma^T \Phi(s_T) \to 0$. Hence $Q'^*(s, a) = Q^*(s, a) - \Phi(s)$ and the argmax over actions, hence the optimal policy, is unchanged [26]. Bonuses that are not potential differences, such as the progress and waypoint bonuses of Sections 8.2 and 10.2, do not have this property. Their effect must be measured on the task endpoint, as Section 9 does.

Partial observability and belief

With observation $o \sim O(\cdot|s)$, the sufficient statistic is the belief, updated by

$$b'(s') \propto O(o \mid s') \int p(s' \mid s, a)\, b(s)\, ds. \tag{B6.11}$$

For linear Gaussian models, (B6.11) is exactly the Kalman filter of Box 5.1. Feeding the filtered position and its standard deviations to the policy (groups 22–26 of Table 10.1) is therefore a Gaussian belief-state approximation of the POMDP.

Model-based reference for the pilot tasks

Solving (7.1) for one plane gives $\dot{\xi} = A\xi + BF$ with $\xi = (x, \dot{x}, \theta, \dot{\theta})$ and $\Delta = (M + m)(I_c + ml^2) - m^2 l^2$:

$$\begin{aligned} \ddot{x} &= \frac{(I_c + ml^2)F - m^2 g l^2 \theta}{\Delta}, \\ \ddot{\theta} &= \frac{mgl(M + m)\theta - mlF}{\Delta}. \end{aligned} \tag{B6.12}$$

A has eigenvalues $0, 0, \pm\lambda_u$ with $\lambda_u = \sqrt{mgl(M + m)/\Delta}$, and $[B, AB, A^2B, A^3B]$ has full rank. The upright equilibrium is therefore unstable but controllable, and an LQR on the linearisation is the natural classical baseline for the CartPole pilot. For the planar 2R arm, a uniform joint sampling density maps to a Cartesian density proportional to $1/|\det J| = 1/(l_1 l_2 |\sin q_2|)$. This accumulates points near the outer ($q_2 = 0$) and inner ($q_2 = \pi$) rings, which explains the concentrations of Figure 7.3.

Evaluating stochastic training

A trained policy is one draw from the distribution induced by the seed, the replay order and the curriculum. With R independent training runs and final scores $X_{(1)} \leq \cdots \leq X_{(R)}$, the interquartile mean

$$\mathrm{IQM} = \frac{2}{R} \sum_{r=R/4+1}^{3R/4} X_{(r)} \tag{B6.13}$$

is robust to a single diverged run. A bootstrap over runs gives its interval [31]. Seed-to-seed variability is a variance component distinct from the path-to-path variability analysed in Box 9.1. With one released checkpoint per method, as in Sections 8–10, it cannot be identified [30]. Every comparison in this report is therefore conditional on the released actors.

7. Preliminary experiments and target reaching

7.1 Two-axis inverted pendulum and planar arm

The pilot used PyBullet and Gymnasium to test actuation, episode termination and learning callbacks on a pole stabilisation problem [2, 3, 33, 34]. A spherical attachment lets the pole tilt in two horizontal directions. For small tilts and a symmetric cart, the two planes can be analysed locally using the same linearisation. With cart mass M, pole mass m, centre-of-mass distance l, pole inertia I_c about that centre, horizontal cart coordinate x, tilt θ and applied force F in one plane:

$$\begin{aligned}(M+m)\ddot{x}+ml\ddot{\theta}&=F,\\ ml\ddot{x}+(I_c+ml^2)\ddot{\theta}-mgl\theta&=0.\end{aligned} \tag{7.1}$$

The sign convention chooses positive tilt as a displacement in positive x from the upright position. The second equation has an unstable gravitational term. Solving the 2× 2 mass matrix gives a local state model in (x,ẋ,θ,θ̇). Large angles, finite cart travel and coupled 3-D effects belong to the simulator. In the reported trials the episode length is capped at 500 steps, and successful trials reach the horizon without falling or leaving the allowed zone. Both methods approach the ceiling, with SAC showing a faster initial rise in the plotted run. Seed-resolved logs, exact reward, sample counts and timestep were unavailable; the figure supports a qualitative pilot comparison.

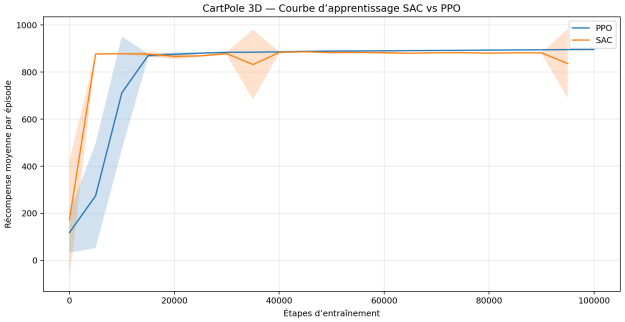


Figure 7.1 – Original two-axis CartPole learning curves from the final project presentation [3, slide 11].

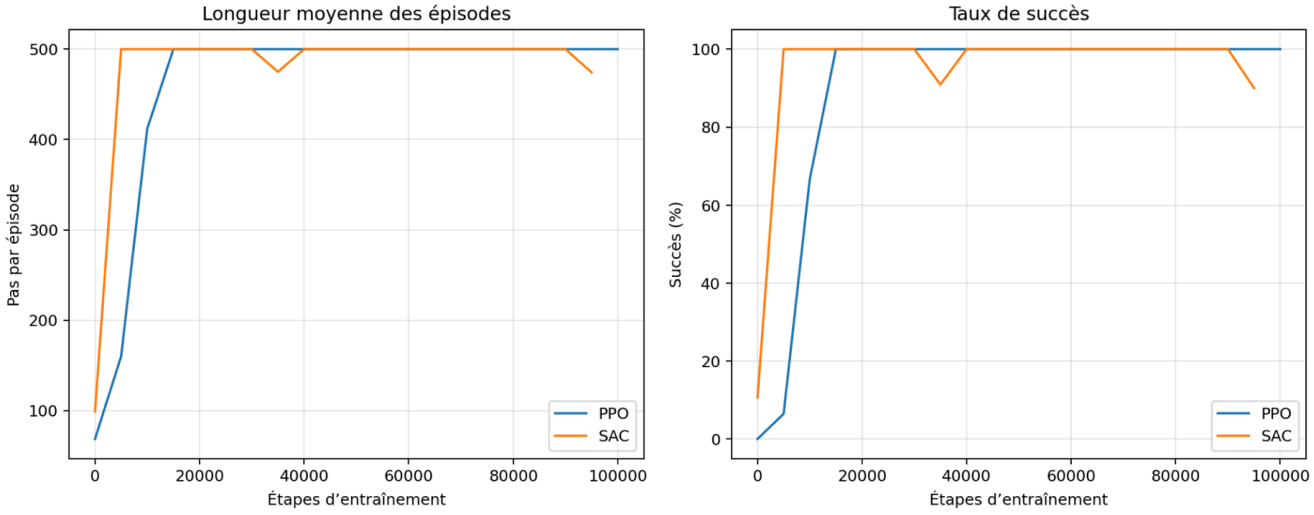


Figure 7.2 – Original episode duration and success plots [3, slide 12]. The two curves approach the 500-step ceiling and reported 100% success in this pilot.

The planar RR example isolates inverse kinematics. For link lengths l_1=10, l_2=5 in the project's plotting units and $q_2 \in [0,\pi]$,

$$\begin{aligned}x&=l_1\cos q_1+l_2\cos(q_1+q_2),\\ y&=l_1\sin q_1+l_2\sin(q_1+q_2),\\ r^2&=l_1^2+l_2^2+2l_1l_2\cos q_2.\end{aligned} \tag{7.2}$$

The annulus is 5≤ r≤15. At either boundary the inverse is singular. The permitted elbow branch gives

$$q_2 = \arccos\frac{r^2 - l_1^2 - l_2^2}{2 l_1 l_2}, \quad q_1 = \operatorname{atan2}(y, x) - \operatorname{atan2}(l_2 \sin q_2, l_1 + l_2 \cos q_2). \tag{7.3}$$

The original network regresses sine and cosine of each angle from the target coordinates. The representation avoids the discontinuous jump at ±π. Recover each angle by atan2(sin q̂,cos q̂) and assess reconstruction both on the circle and in Cartesian space. A uniform sweep in joint angles is not uniform over workspace area because the Jacobian determinant is $l_1 l_2 \sin q_2$. The plotted concentrations near the outer and inner rings are consistent with this geometric effect. The demonstration trains from labelled inverse kinematics generated by FK; it serves as a supervised regression baseline for the project.

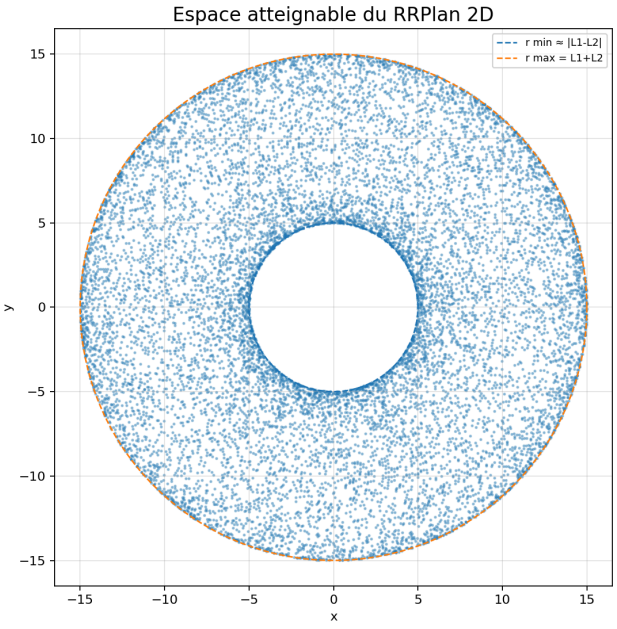


Figure 7.3 – Original 2R reachable-point scatter [3, slide 8]. Curve density reflects the sampling measure of the source joint grid.

7.2 Reaching in a 180° sector

The March report and the current comparison script specify 100,000 training steps for each of five seeds per algorithm, one common test set of 50 targets per trained policy, and a 5 mm threshold [1, 3, 6]. The evaluation compares distance to a target, steps, a command-based effort proxy and tolerance success. For two policies tested on identical targets, uncertainty over training requires repeated training seeds. Using the same targets supports pairing within each training comparison. Evaluating one policy many times at the same targets does not increase the number of independent trained policies.

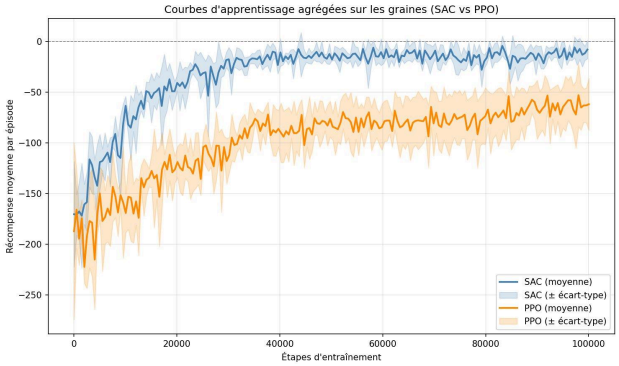


Figure 7.4 – Reported SAC and PPO reaching histories under the nominal 100,000-step budget [3, slide 17]. Bands in the presentation are between-seed dispersions.

At the archived budget, SAC's reported median terminal distance is about 29 mm, and PPO's about 0.4 m. The tolerance sweep gives SAC roughly 13%, 32% and 80% for 5, 20 and 50 mm, respectively. PPO remains below 6% at 50 mm. Thus much of SAC's advantage is in coarse acquisition, with most episodes still outside the 5 mm tolerance at this training budget. The historical figures and available experiment code provide the evidence here. Training logs and per-seed target rows for these exact curves are missing from the cited public repository. The median captions and plot point counts should not be pooled as one reconstructed sample. The release README's significance claim depends on those unavailable seed-level outcomes. The manuscript does not assign a new p-value.

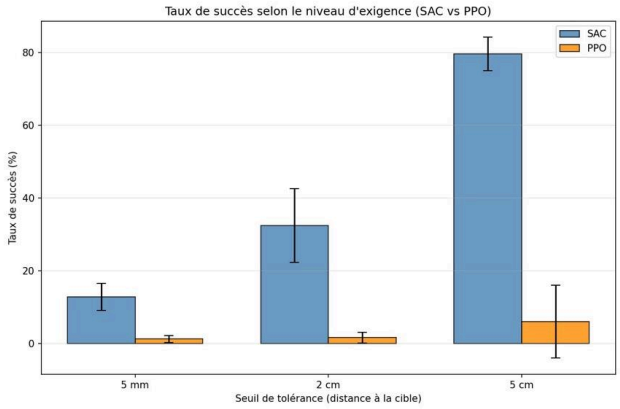


Figure 7.5 – Original tolerance sweep, 5 mm, 20 mm and 50 mm [3, slide 18]. Tolerance success and final distance use the reaching task.

A separate presentation run at 250,000 steps reports 60%, 91% and 98% across those three tolerances on 500 targets. It is consistent with improvement under a larger training budget. The 100,000 and 250,000-step records do not establish an incremental effect for identical seeds and target sets. Timing and computational cost were not established by either rate. Improved terminal accuracy at a larger budget is distinct from the later line-following and pick-and-place tasks.

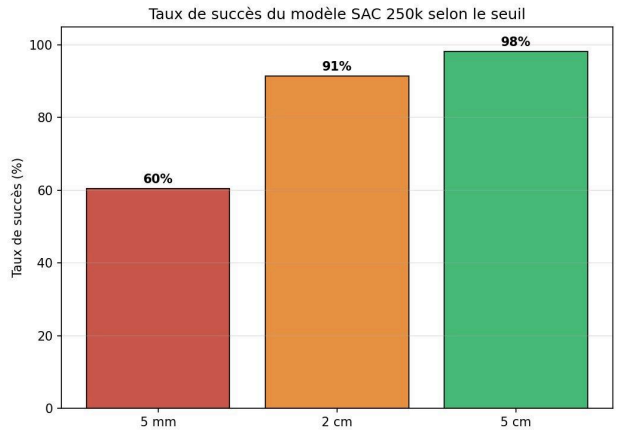


Figure 7.6 – Original reported performance of the longer SAC reaching run on 500 targets [3, slide 19].

7.3 Full workspace, curricula and a fixed obstacle

Later reaching covers a full azimuthal distribution. The extended slides show three projections of a 3,000-target endpoint map with a 50 mm threshold. These are multiple views of one dataset. Clustering of colours reflects endpoint error, sampling density and reachable geometry together. It supplies no complete workspace certificate. A separate initial reaching map reports 12.1 mm median terminal distance and 10% under 5 mm at a stated three million steps. These summaries belong to different protocol or model snapshots from the 180° comparison.

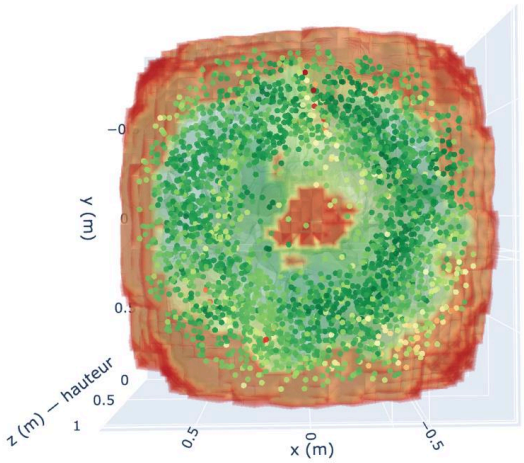


Figure 7.7 – Original plan view of the full-azimuth reaching experiment [3, slide 20]. The gallery preserves the additional views.

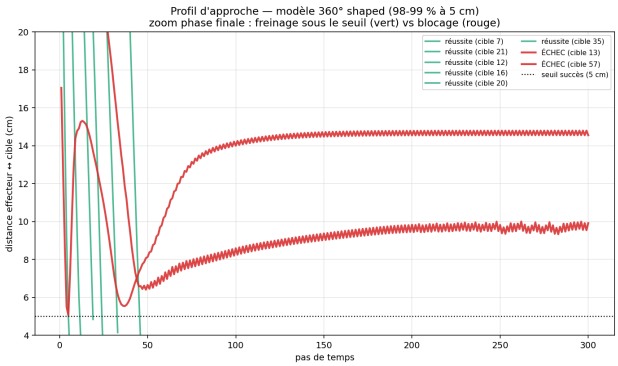


Figure 7.8 – Archived approach trajectories in terminal-distance coordinates [3, slide 21]. The successful and unsuccessful paths are representative examples.

Some goal-approach examples plateau around 10–30 mm in the source. Velocity-limited actions, exploration or damping can contribute, and the plots do not uniquely identify an intrinsic velocity-control limitation. For a discrete positional-error servo $u_k = K\, e_k$ with $e_{k+1} \approx (I - \Delta t K) e_k$, a scalar gain $0 < \Delta t\, K < 2$ is locally stable before clipping in the ideal integrator. This gives a hypothesis for a separately tested end-phase servo. The implemented reaching

family includes dense shaping, tolerance curricula, a goal-conditioned HER variant, and variants in the Stable-Baselines3 and TF-Agents branches [6]. The shared external *ur7e_pybullet* URDF/meshes directory is absent from the pinned public copy. The reported physics results remain archival.

The fixed-obstacle presentation reports about 98% success after 110,000 phase-two steps following reaching adaptation. Its illustrated experiment uses a fixed start and target on opposite sides of a fixed obstacle [3]. Scene views and the distance curve show qualitative avoidance, with no released per-episode table or denominator to assess uncertainty. The separate code branches vary obstacle geometry, start and goal, staged IK/RL responsibility and collision conditions. Pooling them into one rate would mix tasks.

One geometric baseline for a spherical exclusion of centre c and radius r_o is the segment clearance $d=\|a+\beta(b-a)-c\|-r_o$, where $\beta=\mathrm{clip}((c-a)^T(b-a)/\|b-a\|^2,0,1)$. Guard each link and interpolation interval, not merely a sampled TCP position. In articulated space, feasible motion requires respecting joint bounds, swept-link geometry, allowed contact and goal reached. Passing one fixed scenario is evidence of task learning in that scenario. It does not quantify clearance for moving obstacles or guarantee collision-free hardware trajectories.

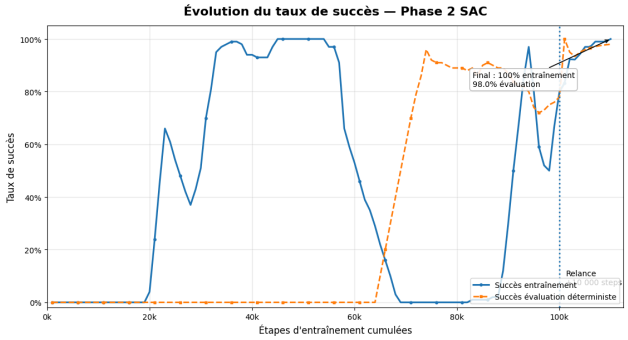


Figure 7.9 – Original fixed-obstacle training/evaluation trace [3, slide 22]. The illustrations and distance signals belong to the staged training experiment.

8. The frozen 29-dimensional line follower

8.1 State and trajectory conventions

The lightweight environment uses a generated URDF, position-only forward kinematics, and one-dimensional ordered progress on a polyline attached to a fixed plane [6, 8]. Five path families appear in the release: segment, quadratic and cubic Bézier, sine and arc. Curves are sampled at 60 vertices and generated with size and reach heuristics. The cubic form, for example, interpolates control points $P_0,\ldots,P_3$:

$$B(t)=(1-t)^3P_0+3t(1-t)^2P_1 \\ +3t^2(1-t)P_2+t^3P_3,\quad t\in[0,1]. \tag{8.1}$$

Equal increments of t generally do not give equal arc spacing. A nearest-vertex distance varies with that spacing even for the same continuous path. Evaluation and direct controller comparisons use the release's own definition. For general path tracking, it is useful to report ordered progress, arc coverage and closest-segment residuals together.

The observation has 29 entries: six normalised joint angles, three TCP coordinates, one progress and one phase coordinate, the three coordinates of the current vertex, and fifteen lookahead coordinates. The five lookahead vertices start at the current cursor and may repeat the endpoint. Thus the current point is also present in the lookahead group. The observation progress is cursor divided by the number of vertices; the published progress statistic divides by that number minus one. A terminal cursor can consequently be encoded below one in the input while reported progress is one. There is no camera image or continuous dynamic state in this experiment.

The action has six real values in $[-1,1]$, mapped to a per-step angular increment with scale 0.05 rad, clipped to the joint bounds. The DLS comparator reads the current-q and TCP components and current target. The actor can also use lookahead and phase. This comparison gives both controllers the same state observations and target sequence, with different use of those inputs. The incremental

kinematic step does not supply a physical clock or a hardware speed.

8.2 Rewards, progression and how early termination arises

Acquisition requires the TCP to enter a 20 mm ball around the first vertex. For tracking, the program uses the smallest Euclidean distance in a cursor-local vertex window and a 25 mm acceptance radius at each target. Its nominal fine-error level is 5 mm and it stops trajectories after a 60 mm local tracking violation or 150 unproductive steps. An 800-step horizon limits the evaluation. The reward gives shaping toward the current point, repeated progress bonuses and a final smoothness/accuracy term. The RMS in the published benchmark uses measured lateral distances for steps already in tracking:

$$E_i = \sqrt{\frac{1}{N_i}\sum_{k=1}^{N_i} e_{ik}^2}, \qquad \overline{E} = \frac{1}{n}\sum_{i=1}^{n} E_i. \tag{8.2}$$

The second expression gives equal weight to each episode. A pooled sample RMS would weight longer trajectories more. Acquisition movements are excluded from E_i; all acquired episodes, including failed endpoints, are retained in the primary comparison.

The source computes distance to the current target before incrementing the cursor. On entry into the penultimate point's tolerance it can increment the cursor to the final index, then use that previous distance in the success test for the final index. For example, two vertices 20 mm apart with the TCP 15 mm behind the penultimate vertex give a penultimate distance of 15 mm and a final distance of 35 mm. The earlier check can terminate even though the endpoint lies outside 25 mm. The original completion flag can also treat stagnation as completion when the episode is terminated and local error is small. The frozen evaluation preserves the archived simulator and separately verifies final-position distance, phase, cursor and maximum tracking error. That audited completion is the performance endpoint used below.

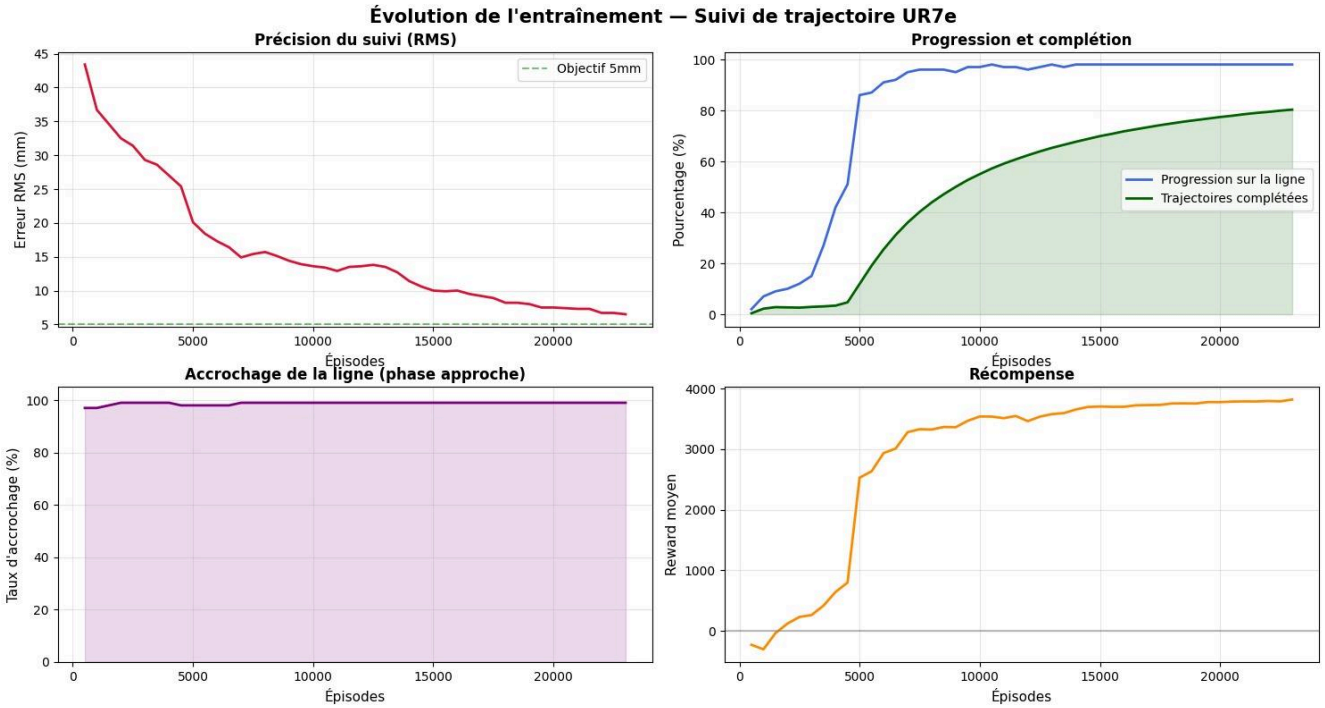


Figure 8.1 – Original simplified line-following training plots [3, slide 24]. The nominal rounded training budget in the slides differs from the precise step count embedded in the retained model.

8.3 Historical stress tests on shapes and lettering

The simplified model historically reported acquisition 100%, completion 97%, mean RMS 5.8 mm and median RMS 4.7 mm over 100 random curves [4, 5, 6]. Curve, timeout and success definitions differ from the strict frozen evaluation. The later audit demonstrates how the original completion flag can be optimistic. It does not provide retroactive strict labels for unreleased historical trajectories.

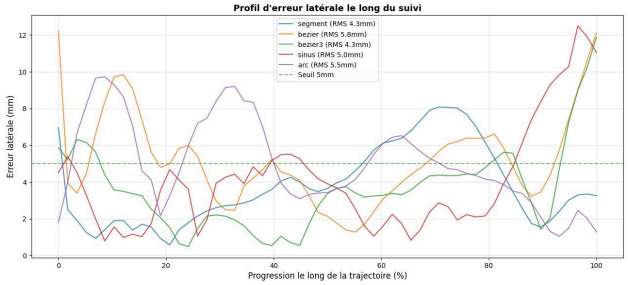


Figure 8.2 – Original representative tracking-error profiles for simple curve shapes [3, slide 26]. The gallery retains the associated histogram, tolerance curve and family summaries.

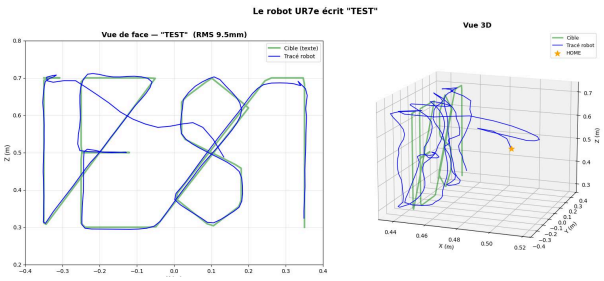


Figure 8.3 – Original trajectory drawing of TEST [3, slide 28]. Reported tracking RMS of 9.5 mm applies to that example.

The typographic demonstrations extend the curve distribution. The TEST example retains some traceable segments. The recorded ARTS, A and B outputs miss sharp corners and show branches or segments inconsistent with the target drawings [3]. A continuous waypoint path approximates pen-up transitions with commanded travel unless the task defines a distinct phase for moving between strokes. A robust benchmark for lettering would represent each continuous stroke, switch the laser state across connector movements and score stroke order, intersections, corners, positive-length missing segments and coverage. Repeated progress on simple random curves does not resolve topology at self-intersections. The original image panel differences are preserved for inspection.

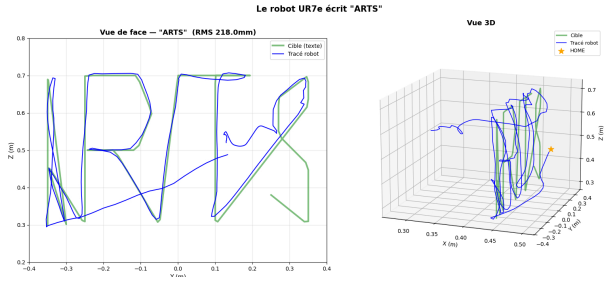


Figure 8.4 – Original ARTS example [3, slide 28]. Stroke topology and transitions differ from the sampled elementary curve families.

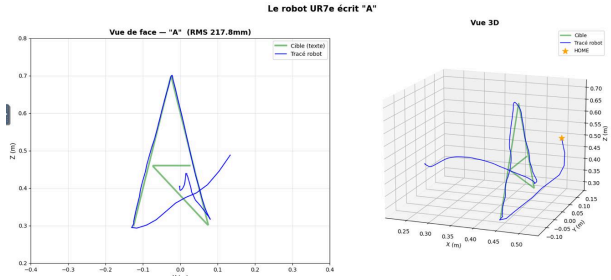


Figure 8.5 – Original letter A experiment [3, slide 29].

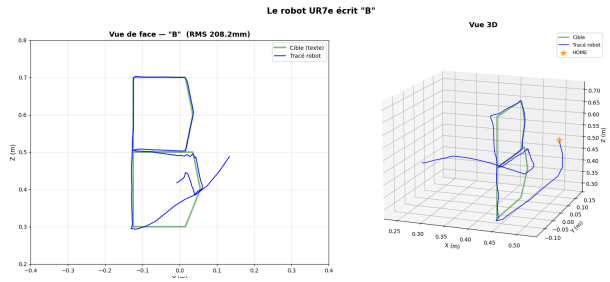


Figure 8.6 – Original letter B experiment [3, slide 29].

9. Paired SAC versus DLS benchmark

9.1 Protocol and reproduction

The frozen release fixes 100 trajectory cases, 20 from each family, with identical seeds and 60 vertices for both controllers. At every step the same pseudorandom stream within each case perturbs six joint angles and the three TCP coordinates. The standard deviation σ is respectively 0, 2 or 5 mrad for joints and 0, 2 or 5 mm for Cartesian

observations. Joint noise is divided by that joint's upper-bound normalising scale in the observed vector. State evolution and evaluation use the actual simulator state. Both controllers are deterministic given their noisy observations. The design has 600 episode rows and 100 distinct trajectories. The original training seed history is unavailable, so the exact overlap with training cases cannot be certified [8, 11, 12].

The fixed classical comparator estimates the 3×6 TCP Jacobian by central differences with $h=10^{-4}$ rad. It takes a single damped position step toward the current vertex:

$$\begin{aligned} \Delta q &= 0.8 J_v^T (J_v J_v^T + 0.02^2 I)^{-1} \\ &\quad (p_{\text{target}} - p_{\text{obs}}), \\ a &= \text{clip}(\Delta q / 0.05, -1, 1). \end{aligned} \tag{9.1}$$

This baseline is deterministic and shared across all six experimental cells. The SAC export applies two affine/ReLU layers and a tanh mean head. The release's parity audit reports maximum absolute differences of 3.70×10^{-6} and 3.35×10^{-6} for per-sample and batched output, over 343 fixtures, compared with Stable-Baselines3 inference [8]. The exported arrays include the deterministic mean path. They omit stochastic sampling and critic training state.

For this manuscript, the released mean actor and the same environment were replayed on CPU using Gymnasium and IKPy. The 600 strict decisions, original flags, acquisition decisions, step counts and tracking sample counts agree with the published episode table. The maximum absolute discrepancies are 0.00225 mm for RMS, 0.0119 mm for final distance and 0.0146 mm for maximum tracking error. These are small numerical differences across runtime/inference paths, not claims about reproducibility of policy training. The script and matched output are embedded with the report.

9.2 Primary results and paired uncertainty

The primary threshold is 25 mm at the final vertex, with phase equal to tracking, the final cursor reached, a terminal stop and no recorded lateral error above 60 mm. In all cells both controllers acquire the line, reach the final cursor and raise the original flag on all 100 cases. The corrected endpoint condition changes 7–13 labels per cell. The observed strict failures are endpoint-tolerance failures. A final cursor and a stop therefore provide weaker evidence than final point distance.

Table 9.1 – Fixed seeded kinematic benchmark. RMS uses all acquired paths. Strict rate is count out of 100 and the Wilson interval is marginal. False positives refer to the inherited flag. σ is in mm and mrad.

σ	Method	Strict	Wilson 95% (%)	Mean RMS (mm)	RMS P50 / P95 (mm)	Flag false +
0	SAC	93	86.3–96.6	5.21	4.93 / 8.14	7
0	DLS	90	82.6–94.5	2.38	2.07 / 4.61	10
2	SAC	92	85.0–95.9	5.75	5.52 / 8.59	8
2	DLS	89	81.4–93.7	3.55	3.25 / 5.52	11
5	SAC	89	81.4–93.7	7.56	7.32 / 9.61	11
5	DLS	87	79.0–92.2	6.50	6.25 / 8.14	13

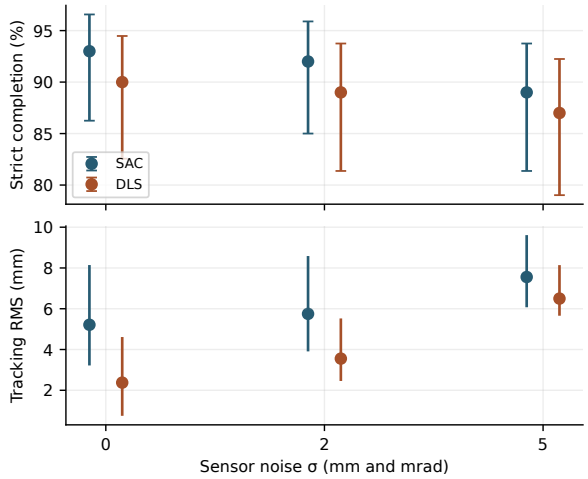


Figure 9.1 – Strict completion count with marginal Wilson intervals and the mean episode tracking RMS with central 90% range of episodes. The RMS bars describe episode spread, not confidence intervals for the means.

No-noise tracking means are 5.214 and 2.376 mm, giving a SAC-minus-DLS mean of 2.838 mm. DLS has smaller tracking RMS on all 100 pairs at zero noise and on 96 and 86 of the pairs at the other noise levels. The endpoint order differs. At zero noise SAC's final distance is smaller in 70 of 100 pairs. Ordered progress, average near-path error and stopping at the endpoint measure different performance aspects.

For a given noise level, let $D_i = Y_{i,SAC} - Y_{i,DLS}$, where Y is a strict indicator or episode RMS. Estimate the marginal difference by $\hat{\Delta} = \sum_i D_i/100$. A stratified bootstrap draws 20 paired indices with replacement within each shape, retains all controller and noise repetitions of each sampled trajectory, and repeats 20,000 times. This quantifies sensitivity to the choice of trajectories under a balanced empirical generator model. It excludes variation between training runs. For completion, an exact conditional sign comparison on discordant pairs is additionally reported. The common joint/position noise across controllers justifies the within-case comparison, not treating the six cells as independent samples.

Table 9.2 – Within-trajectory contrasts, SAC minus DLS. Brackets give the 95% stratified bootstrap interval. σ is in mm and mrad.

σ	SAC only	DLS only	Exact p	Δ completion (pp)	Δ RMS (mm)
0	3	0	0.250	3 [0, 7]	2.84 [2.58, 3.10]
2	3	0	0.250	3 [0, 7]	2.20 [1.96, 2.44]
5	5	3	0.727	2 [−3, 7]	1.06 [0.85, 1.27]

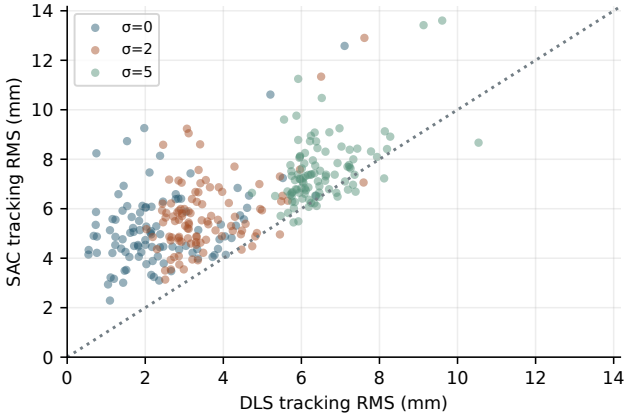


Figure 9.2 – Paired episode RMS. Points above the diagonal favour DLS in lateral tracking. The origin and physical unit are identical on both axes.

No-noise completion differs by three pairs, all favouring SAC. Its exact two-sided conditional p-value is 0.25. The radius and the chosen simulator favour some endpoint trajectories, and this small discordant count gives limited information. At 5 mm/5 mrad the 5 SAC-only and 3 DLS-only pairs give 0.727. The difference in mean RMS declines with the perturbation, from 2.838 to 1.058 mm. The 5-to-0 difference in the SAC-minus-DLS gap has a trajectory-bootstrap interval of approximately −1.90 to −1.66 mm. DLS remains more accurate on that noise range, while its RMS increases more sharply with the injected perturbation. Since the comparator explicitly uses noisy joint angles to form the finite-difference Jacobian, the test gives both errors to that function and its current position feedback. This supports a model-specific explanation of loss under noise. A controlled ablation perturbing only one sensor at a time would isolate each contribution.

9.3 Shape, endpoint radius and performance trade-off

Without added noise, segment, quadratic Bézier and arc reach 20/20 strict for both algorithms. The cubic and sine groups account for all completion misses. At the larger perturbation those more complex shapes still show most failures. DLS has smaller mean RMS within each shape and noise cell. The heat map shows that completion margins remain small whereas tracking margins vary appreciably by family and noise.

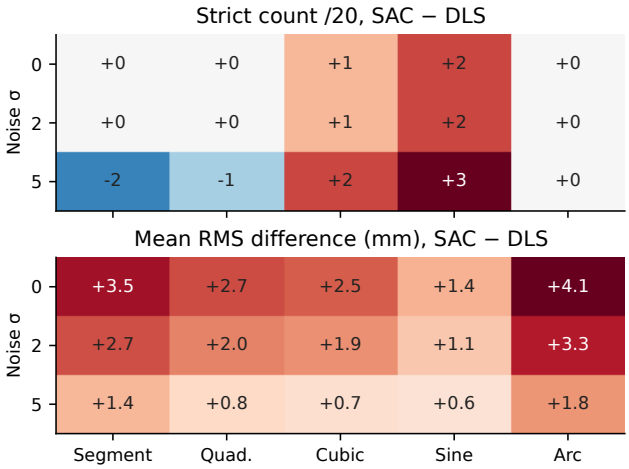


Figure 9.3 – SAC-minus-DLS results for five families. The upper panel gives count difference among 20 cases. The lower panel gives mean tracking RMS difference, in mm.

Changing the endpoint threshold while retaining the archived stop states changes the observed counts. At zero noise and 10 mm, the terminal distances satisfy the threshold in 57 SAC and 43 DLS cases. At 25 mm those counts rise to 93 and 90. At the largest noise level, the 10 mm counts are 44 and 38. This describes terminal states reached under the original termination mechanism. It does not re-evaluate either policy with a different stopping law or an additional settling phase. Terminal-distance survival curves and near-path RMS should be considered jointly when choosing a controller for a task.

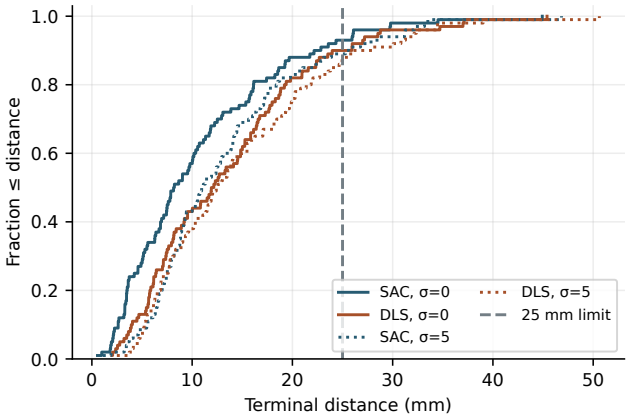


Figure 9.4 – Empirical cumulative distribution of final distance at zero and maximum tested noise. The vertical line marks the strict 25 mm radius.

Original timing fields record inference/comparator compute inside the common benchmark. The median per-episode median was roughly 0.27 ms for SAC and 0.68 ms for the DLS implementation in that release. DLS uses twelve FK calls per central-difference Jacobian, so its latency includes the choice of numeric derivative. The analytic model or a compiled implementation could change this cost. Neither number includes visual, asynchronous message, controller or safety response latency. An inference portability check in the released static Policy Lab compares 18 fixtures and then runs both policies in a browser kinematic reimplementation [10]. It addresses numeric agreement and public inspection; its traces are part of the same frozen experiment.

For this specific position-only task the strongest evidence is lower lateral error with fixed DLS and fewer terminal misses for SAC in the recorded configurations. A fair comparison on the laser task would use the same wall action, calibrated measurement profile, explicit hold at the endpoint, projection error and fault policy for both. Those measurements remain separate from the 3-D fixed waypoint benchmark.

Box 9.1 Statistical inference for paired controller comparisons

Theoretical complement to Sections 9.1–9.3 and Appendix A.2. Exact tests, Holm adjustment, sample sizes and Monte Carlo power computed by theory_ur7e.py.

Score interval for one success rate

For $K \sim \mathrm{Bin}(n, p)$ the score statistic $(\hat{p} - p)/\sqrt{p(1-p)/n}$ is approximately standard normal. Inverting $|\cdot| \le z$ gives the quadratic inequality

$$(\hat{p} - p)^2 \le \frac{z^2}{n}\, p(1-p), \tag{B9.1}$$

whose roots are the Wilson limits (A.2). Unlike the Wald interval $\hat{p} \pm z\sqrt{\hat{p}(1-\hat{p})/n}$, the Wilson interval keeps close to nominal coverage near $p = 1$ and never leaves $[0, 1]$. For 93/100 it gives [86.3, 96.6]%, whereas the Wald interval [88.0, 98.0]% is shifted upward.

Paired binary outcomes

For a given noise level, the two controllers act on the same 100 paths. The outcomes form a 2×2 table with a joint successes, b SAC-only successes, c DLS-only successes and d joint failures. The paired difference and its variance are

$$\hat{\Delta} = \frac{b - c}{n}, \qquad \mathrm{Var}(\hat{\Delta}) = \frac{(p_b + p_c) - (p_b - p_c)^2}{n}. \tag{B9.2}$$

Only discordant pairs carry information. Pairing removes the path-to-path variance shared by both controllers, which is why the comparison of Section 9.2 is made within paths.

McNemar's exact test

Under $H_0 : p_b = p_c$ and conditional on $b + c$, $b \sim \mathrm{Bin}(b + c, \frac{1}{2})$, so

$$p = \min\Big\{1,\; 2\sum_{i=0}^{\min(b,c)} \binom{b+c}{i} 2^{-(b+c)}\Big\}. \tag{B9.3}$$

This is the exact conditional test of Table 9.2: $(b, c) = (3, 0)$ gives $p = 0.25$ at $\sigma = 0$ and 2, and $(5, 3)$ gives $p = 0.727$ at $\sigma = 5$. Holm's step-down correction over the three noise levels multiplies the ordered p-values by $3, 2, 1$ and enforces monotonicity, giving 0.75 for all three. No completion difference is significant, with or without adjustment.

Power and the size of a decisive benchmark

For discordance probabilities p_b and p_c, with $\psi = p_b + p_c$ and $\delta = p_b - p_c$, the number of paired paths needed for a two-sided test of level α with power $1 - \beta$ is approximately (Connor)

$$n \approx \frac{\left(z_{1-\alpha/2}\sqrt{\psi} + z_{1-\beta}\sqrt{\psi - \delta^2}\right)^2}{\delta^2}. \tag{B9.4}$$

A three-point advantage with $p_b = 4\%$ and $p_c = 1\%$ requires $n \approx 434$ paths for 80% power at $\alpha = 0.05$. Monte Carlo simulation of the exact test (20 000 trials per size) gives a power of 9.9% at $n = 100$, 73% at $n = 400$ and 90% at $n = 600$. The 100-path benchmark therefore cannot be expected to resolve completion differences of the size observed. It resolves the RMS contrast easily: 2.84 mm, with a 95% interval of [2.58, 3.10] mm.

Stratified paired bootstrap

With five shape strata of 20 paths, a resample draws 20 path indices with replacement inside each stratum and recomputes the statistic on the pooled resample,

$$\hat{\Delta}^* = \frac{1}{5}\sum_{s=1}^{5}\frac{1}{20}\sum_{j=1}^{20} D_{s,J_{sj}}, \qquad J_{sj} \sim \mathcal{U}\{1, \ldots, 20\}, \tag{B9.5}$$

with $D_{s,j}$ the SAC-minus-DLS difference on path j of stratum s. The 2.5th and 97.5th percentiles of 20 000 draws give the intervals of Table 9.2. Because all controllers and noise levels of a path are resampled together, differences between noise levels, such as the -1.90 to -1.66 mm change of the RMS gap in Section 9.2, keep their within-path correlation. The bootstrap represents uncertainty about which paths the generator happens to produce, not about retraining.

A model for multi-seed designs

A design with several training seeds per method calls for the mixed model

$$Y_{mpsr} = \mu + \tau_m + \beta_s + (\tau\beta)_{ms} + u_p + v_{r(m)} + \varepsilon_{mpsr}, \tag{B9.6}$$

with method m, path p, noise level s and training

run r nested in m. Here $u_p \sim \mathcal{N}(0, \sigma_u^2)$ and $v_{r(m)} \sim \mathcal{N}(0, \sigma_v^2)$. The released benchmark has one run per method, so σ_v^2 is confounded with τ_m: the statement "SAC differs from DLS" is a statement about these two artifacts. With R runs per method, the method contrast has variance of order $2\sigma_v^2/R + 2\sigma_u^2(1-\rho)/n$, where ρ is the within-path correlation of the two methods. This expression shows directly whether additional seeds or additional paths reduce uncertainty more.

Paired continuous outcomes

For the RMS, each path gives a signed difference $D_i = Y_{i,\mathrm{SAC}} - Y_{i,\mathrm{DLS}}$. The distribution-free sign test uses only $N_+ = \#\{D_i > 0\} \sim \mathrm{Bin}(n, \frac{1}{2})$ under H_0, with the same formula as (B9.3). DLS has the lower RMS on 100, 96 and 86 of the 100 pairs (Section 9.2), giving exact two-sided p-values of 1.6×10^{-30}, 6.4×10^{-24} and 8.3×10^{-14}. The Wilcoxon signed-rank statistic $W = \sum_i \mathrm{rank}(|D_i|)\,\mathbb{1}[D_i > 0]$ uses magnitudes as well and is more powerful when differences are symmetric. With differences this consistent, either test leaves no doubt about the direction of the tracking contrast.

Equivalence rather than difference

A non-significant completion difference does not show that the controllers are equivalent. Two one-sided tests (TOST) declare equivalence within a margin $\pm\Delta_0$ at level α when the $(1-2\alpha)$ interval for Δ lies inside $[-\Delta_0, \Delta_0]$:

$$\Delta_0 > \max(|\Delta_{\mathrm{lo}}|, |\Delta_{\mathrm{hi}}|). \tag{B9.7}$$

With the bootstrap interval [0, 7] points at $\sigma = 0$ (Table 9.2), equivalence holds at level 0.025 for a margin of 8 points, but not for 5. The benchmark therefore supports "no completion difference larger than about 8 points", not "no difference".

Censoring and endpoint curves

The terminal-distance survival curves of Figure 9.4 are empirical distribution functions of d_{end}. The Dvoretzky–Kiefer–Wolfowitz inequality gives a simultaneous band of half-width $\sqrt{\ln(2/\alpha)/(2n)}$: 0.136 for $n = 100$ at $\alpha = 0.05$. This band bounds the uncertainty on any threshold count read from those curves, such as the 10 mm and 25 mm counts of Section 9.3. A difference of 14 paths at 10 mm (57 versus 43) is about the size of one band width, so it is suggestive but not conclusive without a paired analysis.

10. Full wall-following architecture and results

10.1 Closed loop, simulator and observation contract

The full ROS 2 model decomposes the loop into fixed-camera observation, target-path geometry, kinematic spot prediction and fusion, SAC wall velocity, damped inverse and joint control. Gazebo provides virtual image/robot data and RViz displays states and traces [6, 35, 36]. Software loops have different clocks: physics ticks, ROS message timestamps and the higher-level action pulse. The observation should pair a joint state and image by time and frame instead of assuming the last messages describe the same pose. In the checked simulation, the controller applies an action over a nominal 25 physics ticks at 250 Hz and then stops or settles before evaluation. Effective task timing also includes messaging and timeout logic.

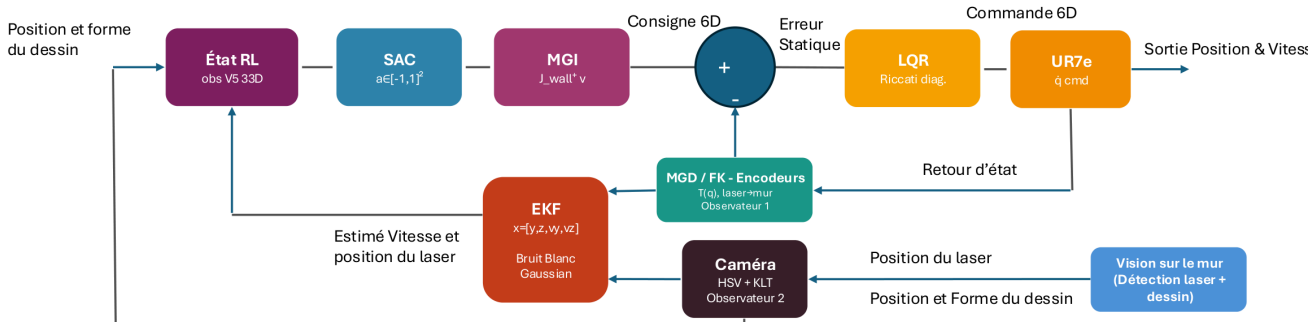


Figure 10.1 – June project architecture [3, slide 33]. The block diagram is an architectural record. Section 3.3 defines the implemented velocity attenuation and Section 5 distinguishes absolute camera and relative line measurements.

Table 10.1 – Feature groups of the complete V5 observation. Indices are zero-based with the upper endpoint excluded.

Indices	Group	Measurement or state
0–6	Normalised joints	Six encoded angular positions
6–9	TCP coordinates	Three forward-kinematics coordinates
9–11	Laser spot	Wall-plane intersection from FK
11–12	On wall	Valid in bounded panel
12–15	Guidance	Two lookahead components and a progress-like guide
15–22	Visual features	Line validity, relative offset, confidence, tangent, coverage and spot flag
22–24	Estimated position	Filtered wall Y and Z
24–26	Position uncertainty	Marginal standard deviations with normalisation
26–29	Conditioning	Wall manipulability and TCP singular-value features
29–31	Previous action	Prior policy output
31–33	Previous wall request	Previous filtered velocity normalised by its cap

The 33 inputs are split into eleven groups. Table 10.1 follows the environment code. Measured features, geometric guidance, two previous velocities or actions, and normalised uncertainty each play different roles. Observation schema and units are part of the checkpoint contract. In the full release, simulated image-based training and testing uses a real-observation mode label even though image and robot sources are virtual. Offline training replaces that stream with features computed from

the known curve and spot. Its KLT-like confidence is exp(−d/0.08) when the dot is on the wall, and its lookahead uses the path model. Thus a high curve-labelled confidence in the offline plots does not measure optical-flow feature survival.

The two action values $a=(a_y,a_z)$ give a requested wall velocity with scaling and temporal smoothing. The checked control code first sets $v_{raw}=0.12\ a/\max(1,\|a\|_2)$ m/s. The filter then uses $v_k=(v_{k-1}+v_{raw})/2$. Nominal damping in that direct map is 0.015 and the joint-speed cap is 0.35 rad/s. The later singularity and guard stages are independent of these nominal inputs. To keep the training state consistent with action smoothing, the preceding normalised action and filtered wall velocity are explicitly included. Integrating a bounded joint velocity through dynamics and using geometric next pose in the offline surrogate produce different state transitions. The simulation report uses both a detailed ROS/Gazebo design and an offline set suitable for quick diagnostics.

10.2 Reward, ordered coverage and timing

The ROS/Gazebo code maintains progress along a projection window of the target polyline. Near the line it rewards incremental ordered progress and waypoint passages; far from it the progress gain is gated. The normalised Huber profile uses a fine penalty near the line and a scaled tail out to 0.50 m. The complete-environment design records a 0.10 m tracking gate, 0.25 per waypoint bonus, 10 completion bonus, 0.20 for loss of vision and action change with factor 0.02 before the relevant scaling. The geometric offline surrogate uses its own finite horizon and reward terms.

A Huber cost with a dimensionless scale $\eta=e/s_{ref}$ can be defined as $\eta^2/2$ for $|\eta|\le 1$, and $|\eta|-1/2$ outside. Its derivative grows near zero and saturates for larger errors. The retained reward code normalises the selected loss over a finite tracking domain. Region and ordered progress gates prevent attaining a high progress score by moving away from the target at excessive speed. Tasks with jumps between strokes need a separate phase. Recent-window RMS helps avoid making recovery impossible after an earlier large transient, while complete-trace RMS remains valuable for reporting.

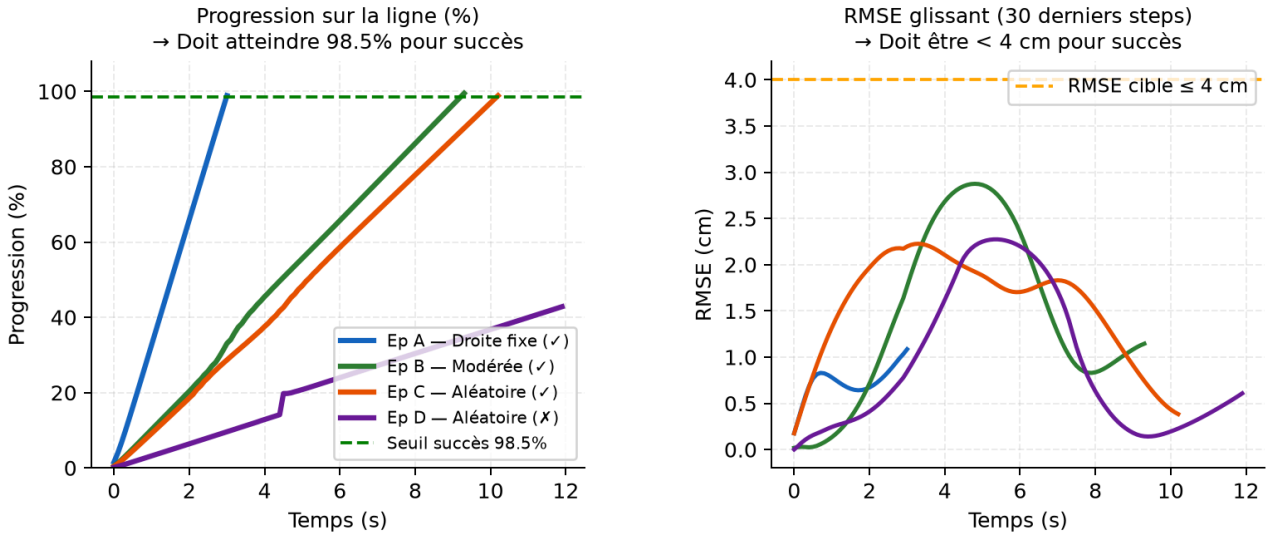


Figure 10.2 – Original progress and recent-error plots [3, slide 34]. Window RMS and ordered coverage have distinct scales and criteria.

The complete-model trainer records a 500,000-step long-run configuration in the slides and carries 256-wide hidden layers. The released 60,000-step offline checkpoint metadata encode a 100,000-entry buffer, start at 1,000 transitions, $\gamma=0.99$, $\tau=0.005$, batch size 256 and automatic entropy. The full ROS training configuration records a 300,000-entry buffer and 5,000 initial

transitions in the slides, with demonstration warm start as another path. Actual checkpoint metadata and training configurations are reported as different artifacts. Later weights or buffer size changes are not treated as historical results for one file.

10.3 The 520-row training record

The public Monitor CSV has 520 complete logged episodes, 59,891 summed episode lengths and wall-time reading 4,293.28 s at the last row [9]. It contains 135 successful terminal episodes (26.0%) and 388 entries of length 120. All 520 rows flag an on-wall terminal state. Terminal on-wall is insufficient by itself to conclude that every intervening frame stayed on the wall. A 60,000-step training counter with 59,891 logged completed-episode steps can leave a final partial episode; the released sources do not map every record to a particular checkpoint transition.

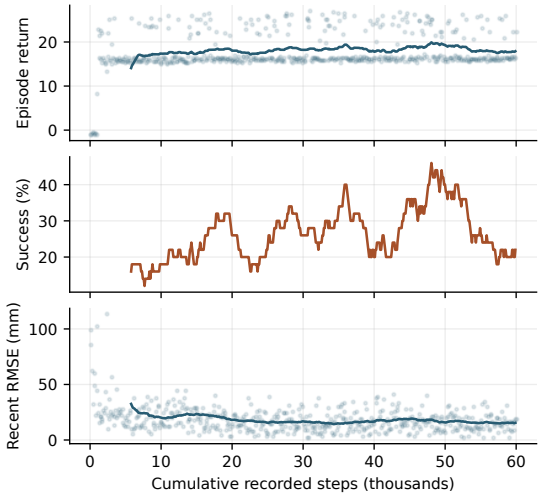


Figure 10.3 – Public training log re-plotted on cumulative recorded steps. Sliding lines use the last 50 episodes. Terminal-window RMSE, raw return and success describe one sequential training run, with changing tasks.

The first 100 logged episodes average 15.88 return, 19 successes, 70.68% progress and 26.38 mm recent RMS at the end. The last 100 average 17.88 return, 24 successes, 75.08% progress and 15.52 mm recent RMS. These descriptive windows give a smaller final error on average while endpoint success remains variable. Episode order and curriculum break the assumptions for treating counts as independent evaluations of one fixed policy. The exact run mix in this train trace cannot be compared with the frozen 100-case strict table as a measure of relative policy performance.

10.4 Independent replay of the ten offline cases

The programme that generates the complete-model plots uses a lightweight geometric *OfflineUR7eEnv*. Its defaults test seed 42 on a straight line, 43–44 at level 0, 45–46 at level 1 and 47–51 at level 2. A deterministic actor was independently reconstructed from the checkpoint tensors and executed with the same ten case definitions. No training or Gazebo process was used. A success requires coverage at least 0.985, recent RMS at or below 40 mm and on-wall status.

Table 10.2 – Independent repeat of the ten offline defaults with the 60,000-step wall actor.

Seed	Level	Steps	Progress (%)	Final recent RMS (mm)	Success
42	−1	31	98.9	10.85	Yes
43	0	77	99.2	13.98	Yes
44	0	64	99.0	11.95	Yes
45	1	94	99.5	11.45	Yes
46	1	120	95.4	11.66	No
47	2	103	98.9	3.83	Yes
48	2	120	83.8	8.01	No
49	2	108	98.7	10.60	Yes
50	2	120	42.8	6.08	No
51	2	112	99.1	22.84	Yes

Seven of ten pass, reproducing the headline count. The three unsuccessful runs reach the 120-step horizon with progress approximately 95.4%, 83.8% and 42.8%. Their final recent RMS is below the threshold, illustrating a progress failure with a small local error. Recent RMS averages 12.21 mm among the seven successes in this replay. The archived README's approximate 8 mm for successful episodes is a separate reported value. The ten replays here use the pinned source and supplied weights, which establish the current summary for those runtime conditions. No raw archived plotting traces or full original environment record resolve the discrepancy in that rounded quantity.

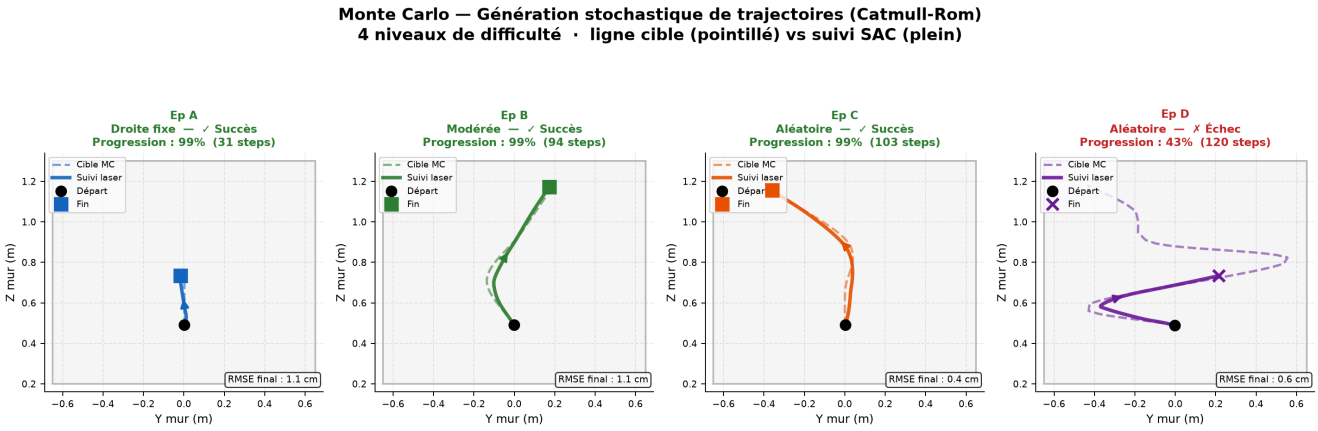


Figure 10.4 – Four archived example trajectories in the extended presentation [3, slide 38]. They communicate qualitatively different path difficulty and stopping modes. The ten-case score is reported separately.

The archived dashboard, Kalman, KLT, geometry and LQR diagnostic figures remain available in the gallery. They use geometric substitutes, simulated noise and a saved actor, as dictated by the published generator. The extended presentation also shows the Gazebo scene. Together these establish a defined simulation architecture and a repeatable offline control experiment. Full closed-loop operation from images in Gazebo or on hardware has a stricter evidence requirement than offline actor replay.

11. Related UR5e pick-and-place simulation

11.1 Plant, policy and object goals

The two notices and extended presentation include a grasp-and-place branch under a UR7e-labelled project title. The released task and historical checkpoint use a UR5e arm with a gripper, as documented in the longer handover notice and the separate repository [5, 7]. The source uses ROS 2 Jazzy and Gazebo Harmonic. Scene variables include robot configuration, cube pose, two platform references, grip state and the target peg. Figure 11.1 preserves the simulated setup. No physical pick-and-place sequence was released.

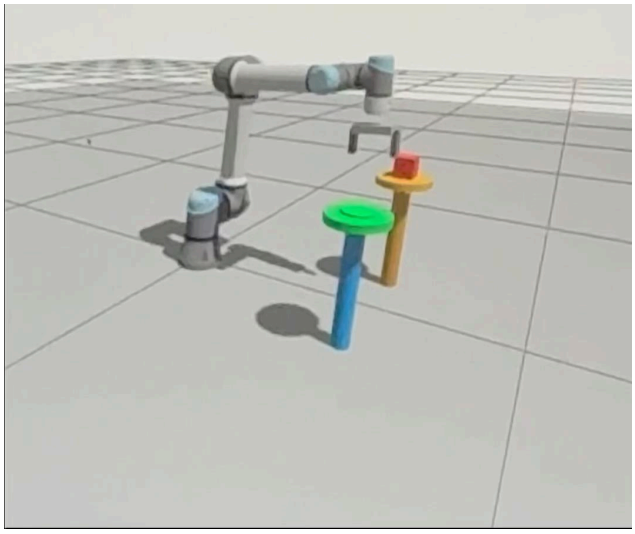


Figure 11.1 – Original pick-and-place Gazebo simulation [3, slide 32]. The checkpoint lineage for this branch is UR5e [5, 7].

The action is four-dimensional, with three normalised requested Cartesian displacements and one gripper channel. The nominal step scale is 0.05 m per Cartesian coordinate. A DLS conversion maps position increments to a bounded joint velocity pulse; the environment integrates motion and can attempt to attach the cube. The grip channel requests a grasp above 0.3 and a voluntary release below −0.85 in the inspected step logic. The current documentation string still mentions −0.3, a value superseded by the code. The nominal grasp controller also checks alignment within an xy radius of 20 mm and a TCP-to-cube vertical range 83–110 mm. These scripted conditions, the simulated detachable joint and Gazebo contacts shape the physical feasibility of the task.

The observation dictionary has a 19-dimensional generic vector, three achieved coordinates and three desired coordinates. The generic group consists of six joint positions, XYZ of the TCP, cube relative to TCP, a gripper-state indicator, and the two 3-D platform positions. The achieved target is the cube centre. Desired coordinates are fixed by the target peg. Training can perturb joint coordinates (2 mrad), cube coordinates (3 mm) and applied displacement (1 mm) in the simulation code. These are scenario settings. They do not estimate encoder quantisation or vision noise of an actual cell.

11.2 Sparse return and an executable relabelling example

The environment gives zero inside or on a 50 mm goal ball and −1 elsewhere. It ordinarily ends on first entrance or at a 60-step time limit. If success first occurs after T failed hypothetical rewards of −1, the previous cumulative discounted value is $-(1-\gamma^{T})/(1-\gamma)$. That structure favours faster success. A fall branch which resets the object within the episode illustrates why termination logic matters. Ending a negative-reward episode early on an undesirable event would sometimes reduce the total penalty relative to continuing.

$$G_0 = -\frac{1-\gamma^{T}}{1-\gamma} \quad \text{for } r_0 = \cdots = r_{T-1} = -1. \tag{11.1}$$

The notebook [13] constructs seven illustrative 3-D positions. Against an original goal (0.35,0.10,0) m every example reward is −1. Relabelling the last achieved position (0.18,0.07,0) m gives zero at the last two points. The penultimate Euclidean distance is about 31.6 mm and the point before that is about 63.2 mm away. The equality threshold tests also verify reward zero at 40 and 50 mm and −1 at 60 mm. The executed demonstration uses synthetic coordinates; the chart belongs to the tutorial, with no inference about manipulation performance.

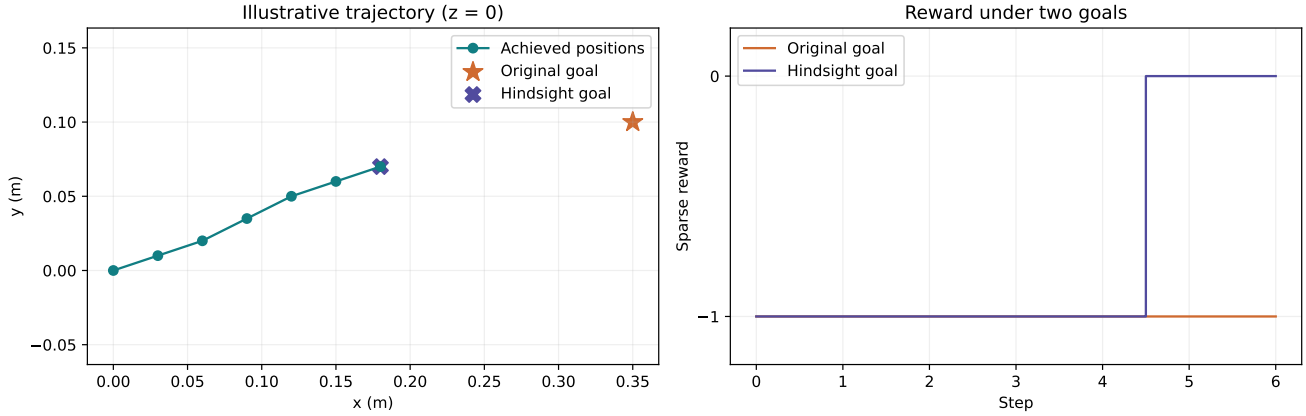


Figure 11.2 – Re-generated figure from the project's public Kaggle tutorial [13]. The same synthetic trajectory is scored with the original and hindsight goals.

The training configuration uses SAC, future-goal HER with four sampled goals, and up to 60 scripted episodes for warm start. It introduces guided resets and progressively expands uniform cube-position perturbation in the plane from ±2 cm to ±8 cm. A step count from a demonstration or guided-reset prefix should be logged separately if computing sample efficiency. Curriculum can change average return even with unchanged policy.

11.3 Reading the manipulation curves

The source reports about 75% success during training with position variation up to ±8 cm. The slide curves illustrate training success, return, durations and auxiliary measurements. The public checkpoint file is labelled 45,000 steps. Its metadata encode automatic entropy, whereas one current trainer setting fixes α=0.05 for fresh training. The saved state thus cannot be identified with every current trainer option. The reported rate is an in-training summary from [7], without a released held-out series or uncertainty on object placement.

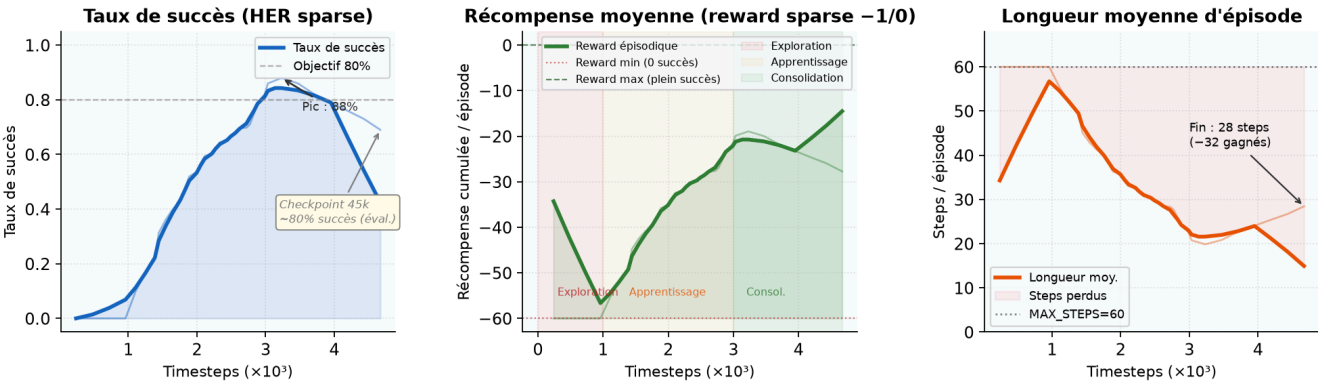


Figure 11.3 – Archived manipulation training curves [3, slide 30]. Task and checkpoints correspond to the simulated pick-and-place branch.

A goal-threshold flag is computed before the environment scripts release and retreat. The final settled cube position and its orientation after release are separate outcomes. A tighter manipulation study would therefore measure approach, successful grip, transport, release, retained placement and time in phase, with terminal geometry evaluated after settling. Grasp rules and the scripted release must remain matched for baselines. None of the existing figures establishes a transfer success rate for real hardware.

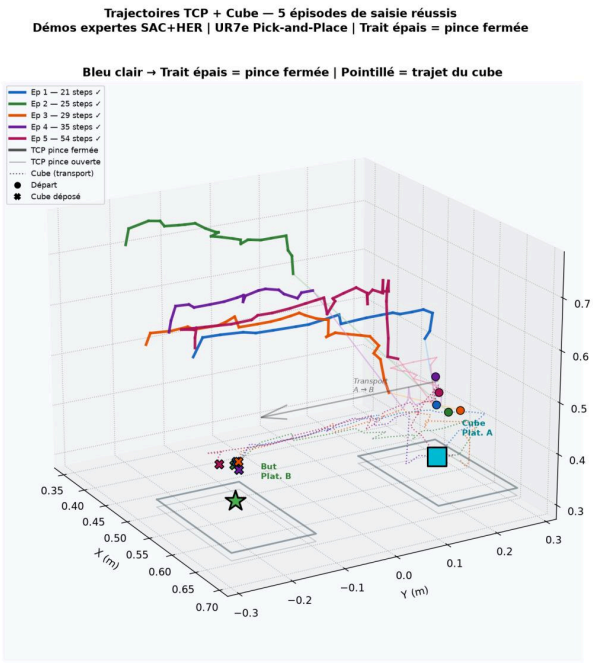


Figure 11.4 – Original example object and TCP trajectories [3, slide 32]. They illustrate the approach and transport phases.

12. Real-robot integration and transfer limit

12.1 Coordinate registration and available observations

The real branch organises an independent ROS 2 driver, a fixed camera, the laser fixture on the tool, homography estimation, observers, a policy runner and diagnostic overlays for geometry [4–6, 37]. The team extracted factory kinematic calibration and confirmed recognition by the robot driver. This supports the transform from joint angles to the tool used by that driver. Accuracy on the wall also requires the flange-to-laser transform, the wall plane and the camera-to-wall map. A calibration test should verify them separately with independent target points. It should record which frame and timestamp each estimate represents. Real-robot state and scene descriptions include parameters that belong to one installation.

The laboratory notices document simultaneous curve and laser detection at rest and include the original camera capture. The moving red spot frequently disappears from the monitored overlay. When a valid camera spot is unavailable or stale, the full runner can inhibit further commanded action through its observation and preflight checks. This behaviour prevents the incomplete perception pipeline from supporting an evidenced end-to-end trajectory. No run in the packet meets a repeatable hardware line-following endpoint.

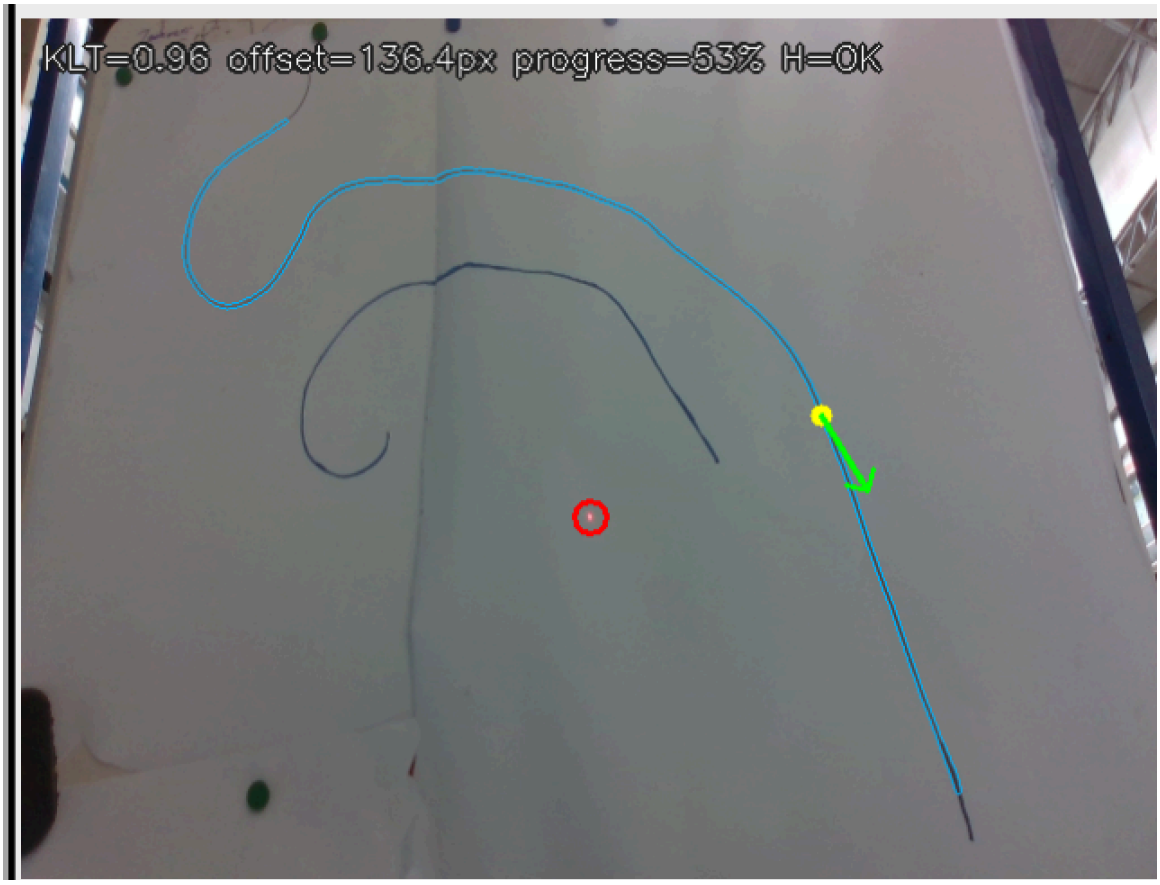


Figure 12.1 – Original stationary setup with detected line and detected laser marker [4, 5]. It documents availability at rest. The notices describe intermittent loss during motion.

12.2 Timing and plausible failure mechanisms

The physical display combines line KLT confidence, colour-based red-spot selection, camera projection, FK and filter estimates. The source image alone cannot identify which detector field failed first, whether a timestamp expired, or how fast the arm was moving. Potential mechanisms include insufficient red value after exposure compensation, weak contrast, blur, reflections, occlusion, a poorly conditioned homography or disagreement between FK and visual spot. Each one implies a different controlled comparison. The synthetic image test gives an example of exposure-induced failure without demonstrating the cause of the laboratory observation.

For time offset δt and local spot speed $\|s\|$, the leading metric mismatch is $\|s\|\,|\delta t|$. A synthetic value of 0.10 m/s with 80 ms mismatch gives 8 mm even when coordinate transforms are correct. A systematic period of laser absence cannot be repaired by treating stale coordinates as fresh or forcing the filter covariance small. The design requires unused measurements to remain flagged with their reason and time. An observer can predict through short missing intervals with increasing uncertainty, and a task-level decision can require hold or reacquisition. These are model and software design choices, distinct from independently rated robot safety functions.

Box 12.1 Sim-to-real error budget, timing and transfer models

Theoretical complement to Sections 3.1, 4.2 and 12. Budget values computed by theory_ur7e.py from the sensitivities quoted in the text; they are modelled, not measured.

First-order error model of the wall spot

Write the spot actually produced and the spot the software believes as $s = \Phi(q, \theta, t)$, with θ the geometric parameters (beam origin and direction, plane, camera map) and t the time stamp. To first order,

$$\delta s = \Phi_q \, \delta q + \Phi_\theta \, \delta\theta + \dot{s} \, \delta t, \tag{B12.1}$$

$$\Sigma_s \simeq \Phi_q \Sigma_q \Phi_q^\mathsf{T} + \Phi_\theta \Sigma_\theta \Phi_\theta^\mathsf{T} + \|\dot{s}\|^2 \sigma_t^2, \tag{B12.2}$$

when the three sources are independent. The individual sensitivities are those already derived in the report: $\ell \, \delta\theta / |\cos\psi|$ for a beam tilt at travel ℓ (Section 3.1), $\delta c \, \tan\psi$ for a plane offset, the homography mapping error of Box 4.1, $\|\dot{s}\|\delta t$ for a time-stamp mismatch (Section 12.2), and the ISO 9283 repeatability for δq (Section 2.1).

Budget

For a 0.1° tilt at 0.5 m, a 1 mm plane offset, the 25-fiducial homography (1.90 mm), a 20 ms mismatch at 0.1 m/s and 0.03 mm repeatability, the root-sum-square (RSS) error at incidence ψ is:

Term (mm)	$\psi = 0°$	30°	60°
Beam tilt	0.87	1.01	1.75
Plane offset	0	0.58	1.73
Homography	1.90	1.90	1.90
Timing, 20 ms	2.00	2.00	2.00
Repeatability	0.03	0.03	0.03
RSS	2.89	2.99	3.70
RSS, 80 ms	8.27	8.30	8.58

Two conclusions follow. First, the arm itself is negligible: its repeatability is more than an order of magnitude below every non-zero term. Second, timing dominates as soon as the mismatch reaches the 80 ms of Section 12.2. To keep a 3 mm budget at $\psi = 30°$, the timing term may use at most $\sqrt{3^2 - 4.96} = 2.0$ mm, that is, a synchronisation error below about 20 ms at 0.1 m/s. That requirement can be checked directly on hardware logs.

Estimating and compensating latency

With FK-predicted spots $s^{\mathrm{FK}}(t)$ and camera spots $s^{\mathrm{cam}}(t_k)$, the latency can be estimated by

$$\hat{\tau} = \arg\min_{\tau} \sum_k \left\| s^{\mathrm{cam}}(t_k) - s^{\mathrm{FK}}(t_k - \tau) \right\|^2, \tag{B12.3}$$

which for small offsets is the peak of the cross-correlation of the two velocity signals. A measurement can then be propagated to the actuation time, $\hat{s}(t + \delta) = \hat{s}(t) + J_W \dot{q} \, \delta$ (or by the filter prediction $F(\delta)$). The residual error is about $\frac{1}{2}\|\ddot{s}\|\delta^2 + \delta \, \sigma_{\dot{s}}$, second order in δ instead of first order. Box 3.2 gives the closed-loop reason: removing one period of effective delay restores the stability margin of the DLS servo from 0.18 to 0.60.

Detection as a function of motion

Section 12 reports that the spot is found at rest but lost in motion. A testable model for logged frames is a logistic detection law,

$$\operatorname{logit} P_D = \beta_0 + \beta_1 \|\dot{s}\| \, T_e + \beta_2 \, \mathrm{gain} + \beta_3 \, m_V, \tag{B12.4}$$

where T_e is the exposure time, $\|\dot{s}\|T_e$ the blur length of Box 4.2 and m_V the value margin of Figure B4.1. The coefficients are estimated by maximum likelihood on labelled frames. A significant β_1 with a vanishing effect once m_V is included would confirm the exposure-threshold mechanism identified in simulation (Box 4.2). Tests based on the Fisher information of β (Wald or likelihood ratio) make the hypothesis table of Table 12.1 quantitative.

Transfer gap and domain randomisation

Let $J_{\mathrm{sim}}(\pi)$ and $J_{\mathrm{real}}(\pi)$ be the returns of a policy in two MDPs that share rewards bounded by $r_{\max}$ and whose transition kernels differ by at most $\varepsilon = \max_{s,a} \|P_{\mathrm{real}}(\cdot|s,a) - P_{\mathrm{sim}}(\cdot|s,a)\|_1$. The simulation lemma bounds

$$|J_{\mathrm{real}}(\pi) - J_{\mathrm{sim}}(\pi)| \le \frac{\gamma \, r_{\max} \, \varepsilon}{(1-\gamma)^2}. \tag{B12.5}$$

The factor $1/(1-\gamma)^2$ (here 10^4) explains why small, systematic model differences, such as the timing

and detection effects above, can produce large performance gaps over long horizons. Domain randomisation replaces J_{sim} by an expectation over simulator parameters ξ,

$$\max_{\pi} \ \mathbb{E}_{\xi \sim p(\xi)}[J(\pi;\xi)] \quad \text{or} \quad \max_{\pi} \ \min_{\xi \in \Xi} J(\pi;\xi), \tag{B12.6}$$

the second form being the robust (worst-case) counterpart [38]. Either form is useful only when $p(\xi)$ or Ξ covers the real system. This is the condition stated in Section 12.3: randomisation should be motivated by logs, with the latency, detection and calibration distributions estimated as above.

What the theory asks of the next experiment

The boxes converge on a short list of measurable quantities, each tied to a model in this report:

- the synchronisation error $\hat{\tau}$ (B12.3) and its spread;
- detection margin and P_D versus speed (B12.4, Box 4.2);
- NIS mean and rejection rate on valid frames (Box 5.2);
- homography error at withheld fiducials (Box 4.1);
- the measured gain of the delayed loop against $g^*(d)$ (Box 3.2);
- for learning, several seeds per method and at least about 400 paths (Box 9.1).

Table 12.1 – Measurements that distinguish failure mechanisms at the real integration stage.

Hypothesis	Observable test
Camera dropout	Frame-wise labelled spot and line validity by stationarity, motion and appearance.
Blur or exposure	Spot extent and intensity, shutter/exposure state, and true image displacement.
False localisation	Distance to annotated centroid, false-red association and recovery time.
Registration error	Hold-out wall fiducials, independent beam registration and residual vector field.
Asynchrony	Acquisition timestamps, encoder/vision lag and innovations by speed.
Control integration	Observation gate reasons, shadow-mode outputs and unavailability intervals.

A lab-approved test sequence can compare image-based red spot and line detection on independently annotated visual traces before any policy-controlled movement. Log classified misses, distractor matches, time to reacquisition, consecutive dropout length, FPS, exposure and frame timestamps. Test image-to-wall accuracy at withheld fiducials across the panel and establish any FK drift by an independent reference. The independent check should synchronise camera and encoder streams and avoid reusing correspondences fitted from FK as absolute validation. Shadow-mode output can record planned actions and gate reasons without motion. Only a separate hardware test plan and independently enforced robot measures can establish the full closed-loop safety and efficacy envelope [37]. This project report contains no hardware commissioning instruction.

12.3 Generalisation and experiment design

Simulation can vary appearance, camera pose, delays, beam dispersion, localisation bias and dropout patterns. Such randomisation is helpful only when its distribution is motivated by logs and when the evaluation includes withheld conditions [38]. A scientifically useful next comparison would freeze a wall servo, the current actor with matching checkpoints, and an actor trained with calibrated perturbations. Sensor and method groups would share the same timestamps and wall curves. Report acquisition, ordered completion, final distance, mean and tail tracking error, valid-frame and stop fractions, and confirmation by an independent wall reference. Use condition- and seed-level repeats instead of pooling visually selected demonstrations.

Cross-domain transfer should document three different losses: uncertainty in geometric map $s(q)$, observation distribution change and dynamics mismatch. The first may shift the estimated path even at rest. The second may produce wrong visual guidance or a loss of the admissible observation state. The third changes the realised motion relative to the action. The empirical bottleneck in the reported setup is loss of visual spot tracking during movement. The evidence is insufficient to measure the contribution of each mechanism.

13. Discussion and conclusions

The simulator and source snapshots illustrate why a robot-learning claim needs both a task definition and an evidence level. In the fixed paired benchmark of 100 distinct paths at three noise levels, DLS maintains lower near-path RMS in every tested group. The no-noise mean is 2.38 mm for DLS and 5.21 mm for SAC. SAC completes 93 paths to DLS's 90 at the strict 25 mm radius, with three discordant cases. The paired data do not establish a reliable terminal-completion advantage beyond the tested apparatus. Replay reproduces all 600 endpoint decisions and steps. A separate analysis shows that the inherited cursor flag counts every episode as finished even when the final displacement fails tolerance.

The extended architecture includes laser-plane projection, damped inverse control, scheduled attenuation and filtering. The nominal Jacobians match finite differences and show sensitivity to grazing incidence. Independent synthetic

experiments demonstrate a reduction in held-out homography RMS with more fiducials, exposure sensitivity of spot detection and limited benefit of gating when a systematic FK bias dominates. The complete-model actor passes seven of ten offline cases with geometric visual stand-ins. The public Monitor log shows trends during training for one run. The independent UR5e pick-and-place branch demonstrates a separate goal-conditioned simulation using HER and scripted grasp and release.

The original camera evidence documents detection at rest and missing laser observations during motion. A repeatable real-world closed-loop result and a measured absolute tracking accuracy are still open. The paired benchmark, source provenance and the complete-project diagnostic curves remain part of one manuscript so that every quantitative conclusion can be traced to the relevant experiment.

References

[1] H. Simon, B. Kulik, D. Kumar and K. Belin. *Projet Cobot Mécatronique 2026 — Apprentissage par renforcement.* Intermediate project document, 18 March 2026.

[2] H. Simon, D. Kumar, B. Kulik and K. Belin. *Projet Expertise Mécatronique.* Final presentation, 34-slide PDF, June 2026.

[3] H. Simon, D. Kumar, B. Kulik and K. Belin. *Projet Expertise Mécatronique.* Extended 42-slide PDF export, June 2026.

[4] H. Simon, D. Kumar, B. Kulik and K. Belin. *Notice technique de reprise.* Four-package version, June 2026.

[5] H. Simon, D. Kumar, B. Kulik and K. Belin. *Notice technique de reprise.* Five-package version, June 2026.

[6] D. Kumar. *UR7e Reinforcement Learning Suite.* GitHub, revision 0c26b57, 2026.

[7] D. Kumar. *UR5e Reinforcement Learning Pick and Place.* GitHub, revision f332fc6, 2026.

[8] D. Kumar. *UR7e SAC line following.* Hugging Face model release, revision a00b40d, 2026.

[9] D. Kumar. *UR7e simulated line-following training log.* Hugging Face dataset, revision d21fc5f, 2026.

[10] D. Kumar. *UR7e Policy Lab.* Static policy demonstrator and runtime fixture checks, revision 590673f, 2026.

[11] D. Kumar. *UR7e SAC vs DLS seeded kinematic benchmark.* Kaggle dataset and loaded actor, 2026.

[12] D. Kumar. *UR7e SAC vs DLS reproducible policy benchmark.* Kaggle notebook, 2026.

[13] D. Kumar. *Sparse rewards and HER for UR5e robots.* Kaggle notebook, 2026.

[14] Universal Robots. *UR7e technical specifications.* Product datasheet, October 2026 revision.

[15] K. M. Lynch and F. C. Park. *Modern Robotics: Mechanics, Planning, and Control.* Cambridge University Press, 2017. Companion site.

[16] Y. Nakamura and H. Hanafusa. Inverse kinematic solutions with singularity robustness for robot manipulator control. *Journal of Dynamic Systems, Measurement, and Control*, 108(3), 163–171, 1986. 10.1115/1.3143764.

[17] T. Yoshikawa. Manipulability of robotic mechanisms. *International Journal of Robotics Research*, 4(2), 3–9, 1985. 10.1177/027836498500400201.

[18] D. P. Bertsekas. *Dynamic Programming and Optimal Control.* Vol. I, 4th ed., Athena Scientific, 2017.

[19] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* 2nd ed., Cambridge University Press, 2004. 10.1017/CBO9780511811685.

[20] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11), 1330–1334, 2000. 10.1109/34.888718.

[21] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *IJCAI*, 674–679, 1981. Paper.

[22] J. Shi and C. Tomasi. Good features to track. *CVPR*, 593–600, 1994. 10.1109/CVPR.1994.323794.

[23] J.-Y. Bouguet. *Pyramidal implementation of the Lucas Kanade feature tracker.* Intel technical report, 2001. Technical note.

[24] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1), 35–45, 1960. 10.1115/1.3662552.

[25] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction.* 2nd ed., MIT Press, 2018. Companion site.

[26] A. Y. Ng, D. Harada and S. Russell. Policy invariance under reward transformations: theory and application to reward shaping. *ICML*, 278–287, 1999. Paper.

[27] T. Haarnoja, A. Zhou, P. Abbeel and S. Levine. Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor. *ICML*, PMLR 80, 1861–1870, 2018. Paper.

[28] T. Haarnoja et al. Soft actor-critic algorithms and applications. arXiv:1812.05905, 2018.

[29] J. Schulman, F. Wolski, P. Dhariwal, A. Radford and O. Klimov. Proximal policy optimization algorithms. arXiv:1707.06347, 2017.

[30] P. Henderson et al. Deep reinforcement learning that matters. *AAAI*, 32(1), 2018. 10.1609/aaai.v32i1.11694.

[31] R. Agarwal, M. Schwarzer, P. S. Castro, A. Courville and M. G. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. *NeurIPS*, 34, 29304–29320, 2021. Preprint.

[32] M. Andrychowicz et al. Hindsight experience replay. *NeurIPS*, 30, 2017. Preprint.

[33] M. Towers et al. Gymnasium: a standard interface for reinforcement learning environments. arXiv:2407.17032, 2024.

[34] A. Raffin et al. Stable-Baselines3: reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268), 1–8, 2021. Paper.

[35] S. Macenski, T. Foote, B. Gerkey, C. Lalancette and W. Woodall. Robot Operating System 2: design, architecture, and uses in the wild. *Science Robotics*, 7(66), eabm6074, 2022. 10.1126/scirobotics.abm6074.

[36] N. Koenig and A. Howard. Design and use paradigms for Gazebo, an open-source multi-robot simulator. *IROS*, 3, 2149–2154, 2004. 10.1109/IROS.2004.1389727.

[37] Universal Robots. *UR calibration and ROS 2 driver documentation.* Kinematic calibration package and manufacturer manuals.

[38] X. B. Peng, M. Andrychowicz, W. Zaremba and P. Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. *ICRA*, 3803–3810, 2018. 10.1109/ICRA.2018.8460528.

Appendix A. Procedure and model checks

A.1 Trajectory benchmark procedure

The replay uses the release's order and strict classification. At each step the same functions generate q and Cartesian observation noise for that seed, the controller estimates a bounded increment, and the environment supplies the next configuration. The audit uses true simulated geometry after each update. The numbered operations below specify the algorithm independently of display or language.

```
For each shape and its 20 registered seeds
  Generate 60 nominal points.
  For each noise scale and controller
    Reset to the common joint configuration.
    For each step up to the common horizon
      Perturb only the measured q and TCP.
      Infer the action and step the environment.
      Accumulate errors only after acquisition.
      Exit on a terminal condition or time limit.
    Record endpoint, tracking RMS and both flags.
```

The classification predicate in the code combines four tests:

$$C_i = \mathbf{1}\{\text{terminal}_i,\ \text{phase}_i = 1,\ c_i = N-1,\ d_{i,\text{end}} < 25\ \text{mm},\ \max e_i \le 60\ \text{mm}\}. \tag{A.1}$$

c_i is the last reference cursor. The same 100 cases are used at all noise levels. For success proportion $\hat{p}=k/n$, the Wilson interval solves $(\hat{p}-p)^2 \le z^2\, p(1-p)/n$ for p. Its centre and half-width are

$$c = \frac{\hat{p} + z^2/(2n)}{1 + z^2/n},\qquad h = \frac{z}{1+z^2/n}\sqrt{\frac{\hat{p}(1-\hat{p})}{n} + \frac{z^2}{4n^2}}. \tag{A.2}$$

Those marginal intervals are shown for transparency. The comparison across paired controllers uses the distribution of within-case differences and the 20-case shape strata. The hypotheses and conclusions are specific to the fixed pair of controllers and the case generator.

A.2 Numerical and observational checks

Forward geometry is tested by direct actuation of each coordinate and central finite differences. Beam intersection is checked at a positive nominal travel with a non-grazing denominator; extrapolating the result to nearly parallel rays is unjustified. Normalised homography fitting uses only oriented planar truth and known fiducial assignment. The vision test supplies the true starting pixel to the tracker and uses known synthetic gain, blur and noise. The observer Monte Carlo uses identical generated motion, camera outages and outliers for each filtering strategy.

The UR7e notebook was executed with its published Kaggle dataset [11, 12]. The checksums, total rows and balance assertions pass. Its 18 reference observations agree with the NumPy actor within 1.70×10^{-6} in action. The Policy Lab's separate Node test on the same number of reference fixtures agrees within 9.54×10^{-7} in action and 5.96×10^{-8} m in FK. The two successfully replayed example paths in that test are selected fixtures of the public frontend. Figure A.1 reproduces the notebook's five path examples.

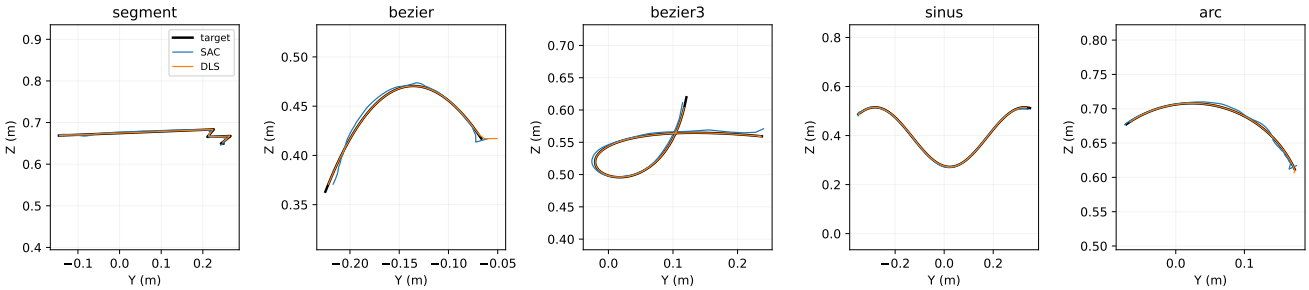


Figure A.1 – Five published trajectory examples regenerated from the benchmark notebook [12]. The plotted traces are sampled after acquisition, and the axis scale belongs to each path.

Appendix B. Original project figures

The following plates retain technical source figures across the intermediate report, both presentations, the simulation repository and the published release. Original annotation remains in its publication language. Panels that recur in two presentations are represented by one technical image when their data are the same. Figures in the main text are cross-referenced by their sources.

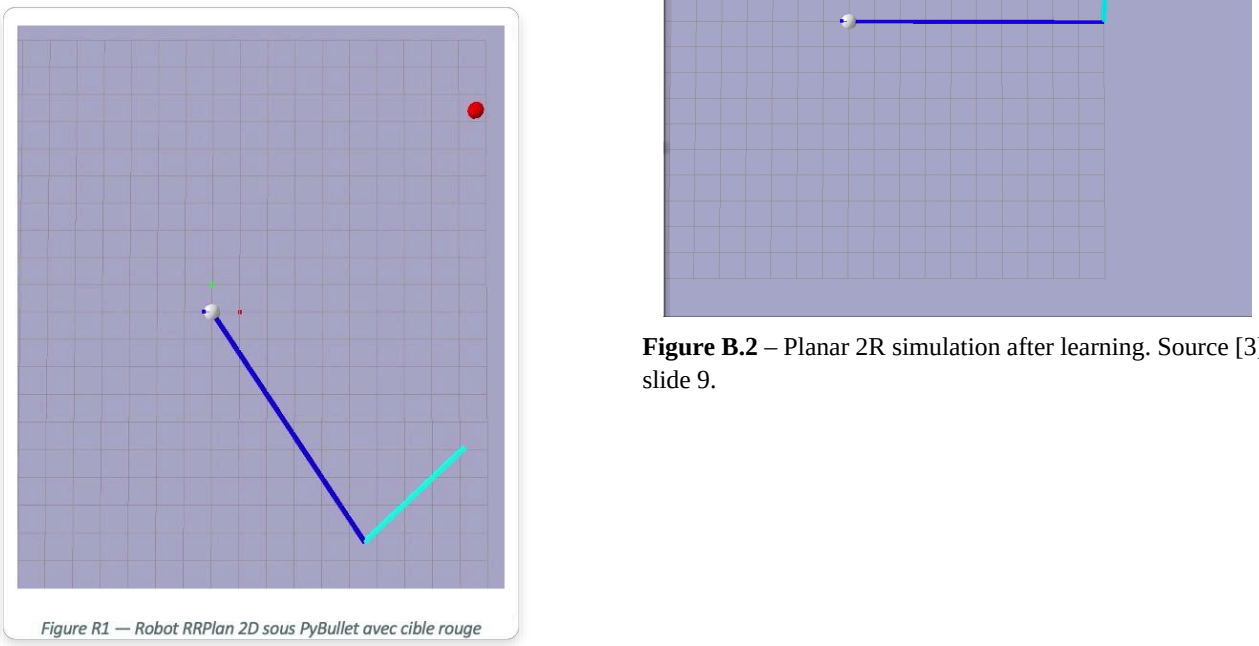


Figure B.1 – Planar 2R inverse-kinematics prototype. Source [3], slide 7.

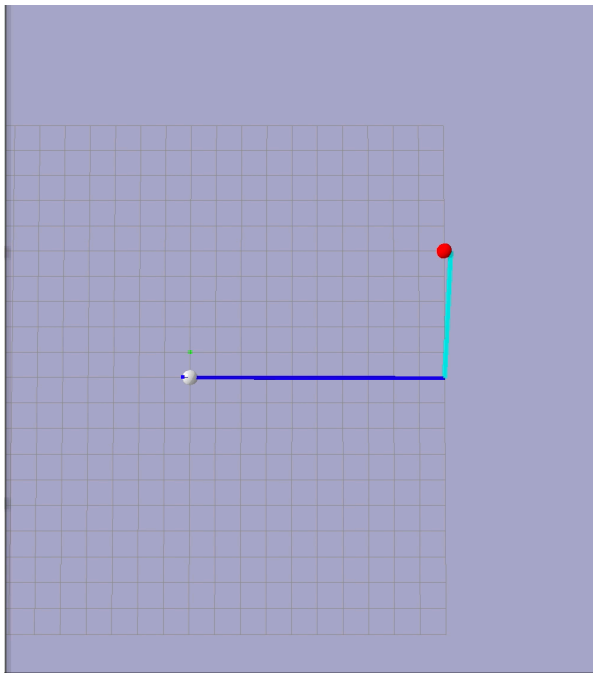


Figure B.2 – Planar 2R simulation after learning. Source [3], slide 9.

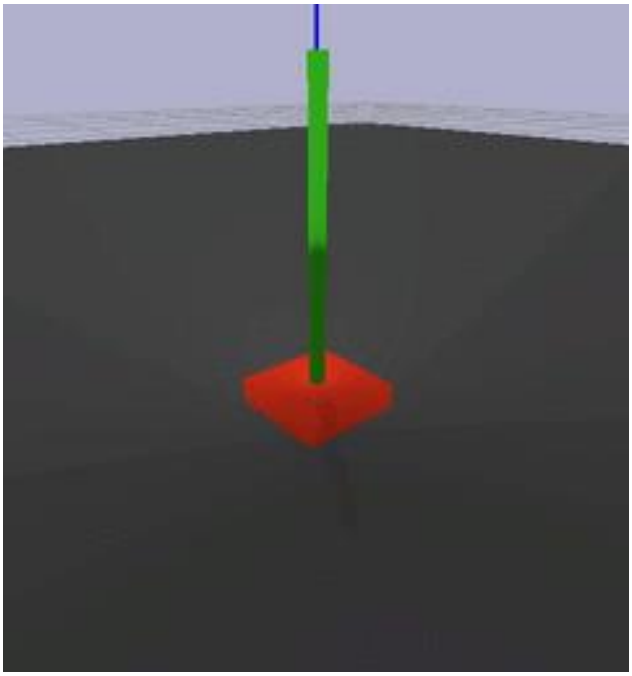


Figure B.3 – Two-axis CartPole simulation. Source [3], slide 10.

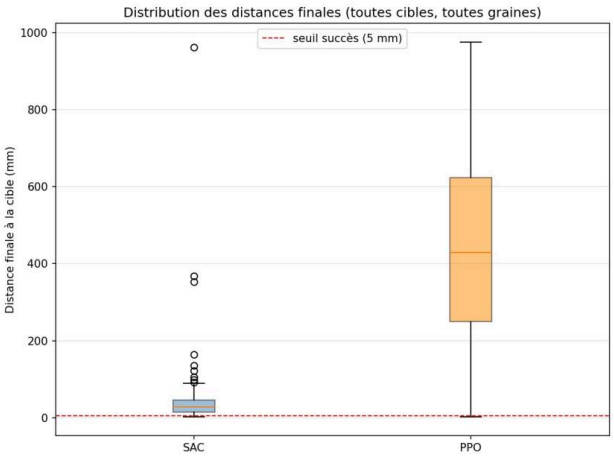


Figure B.4 – Half-space terminal-distance distributions. Source [3], slide 18.

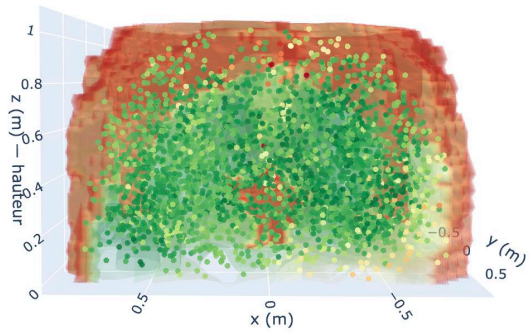


Figure B.5 – Reaching over 360°, side view. Source [3], slide 20.

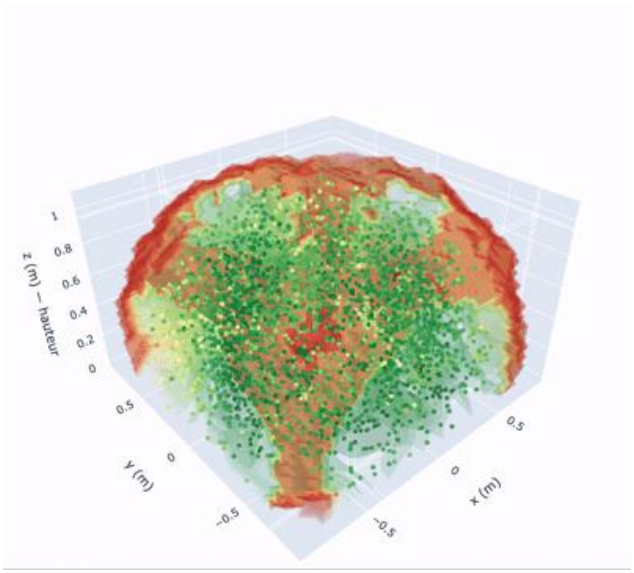


Figure B.6 – Reaching over 360°, global view. Source [3], slide 20.

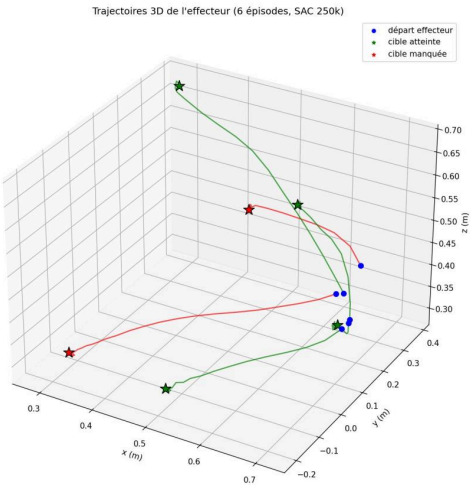


Figure B.7 – Example 3-D reaching paths. Source [3], slide 21.

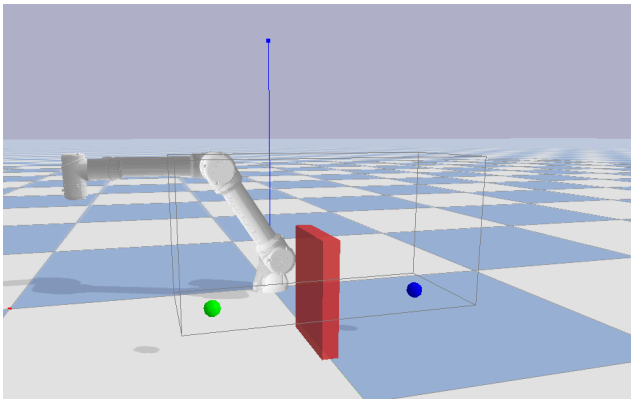


Figure B.8 – Fixed-obstacle test, view A. Source [3], slide 22.

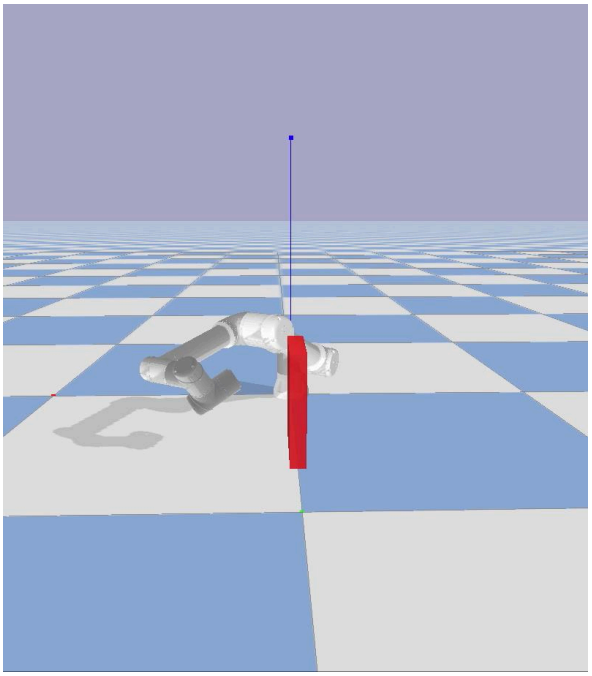


Figure B.9 – Fixed-obstacle test, view B. Source [3], slide 22.

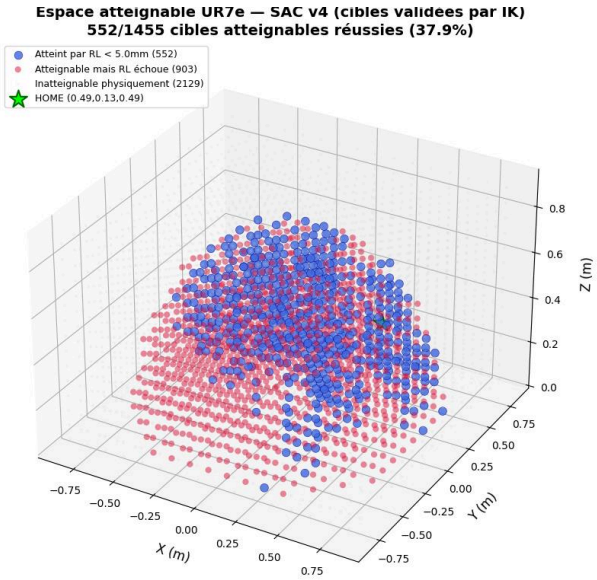


Figure B.10 – Reaching map following a reported three million steps. Source [3], slide 23.

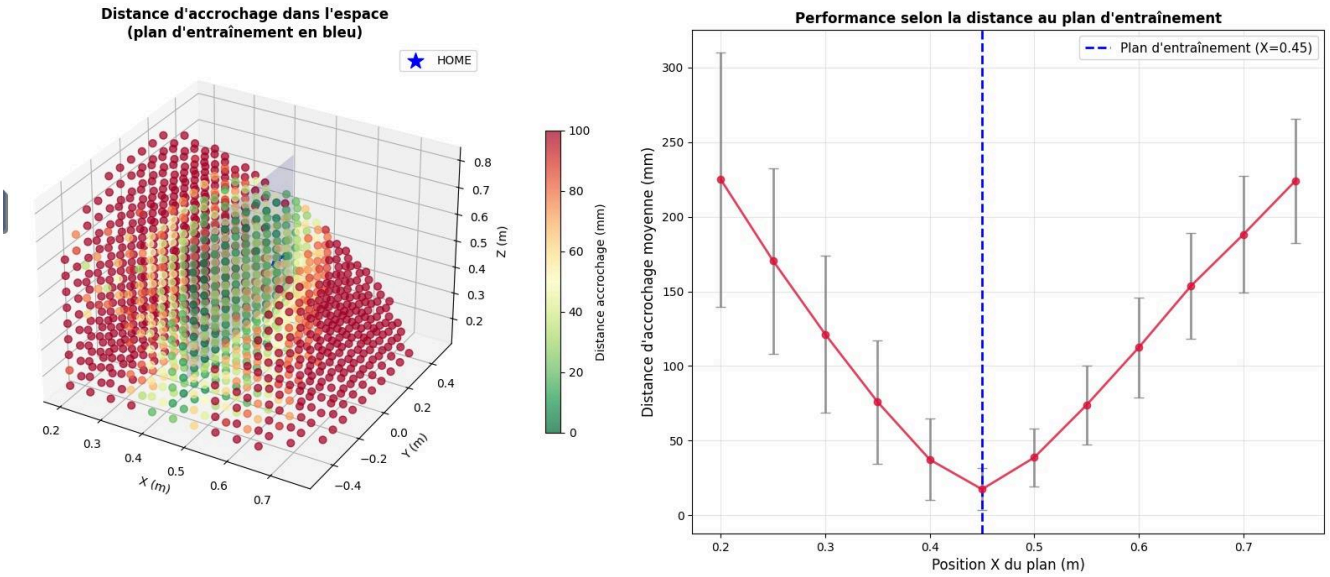


Figure B.11 – Workspace map and spatial variation. Source [3], slide 25.

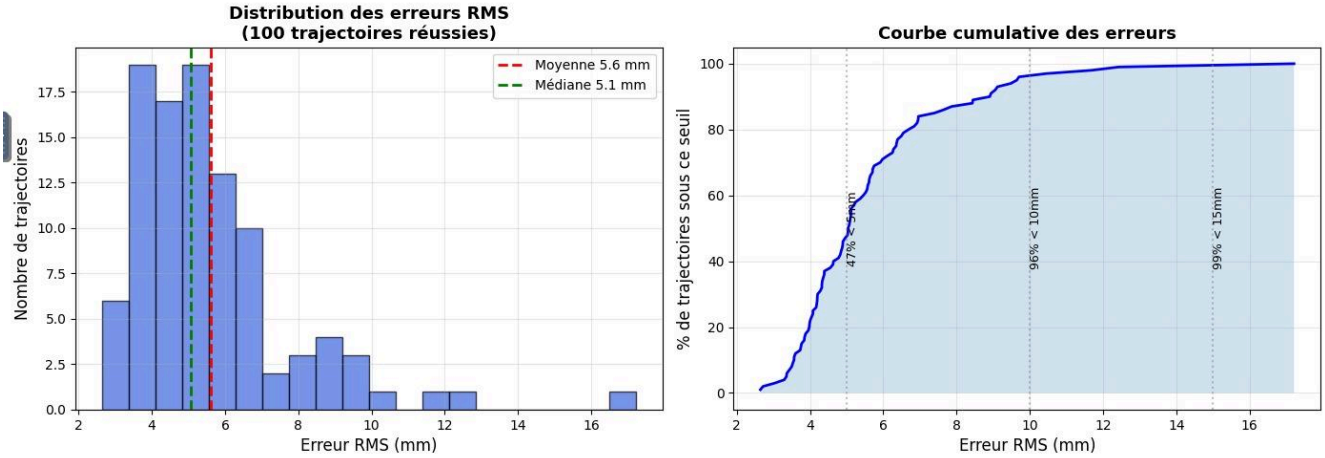


Figure B.12 – Initial simplified model RMS and tolerance curve. Source [3], slide 26.

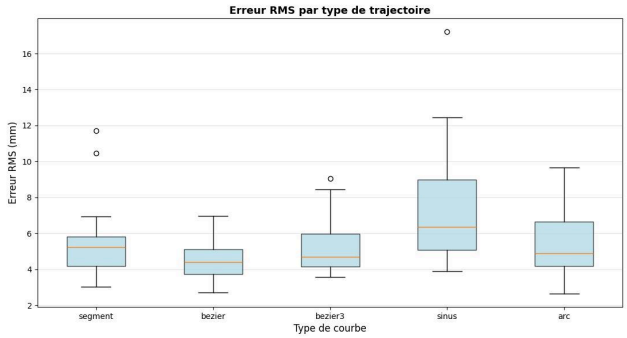


Figure B.13 – Simple-model RMS stratified by curve type. Source [3], slide 26.

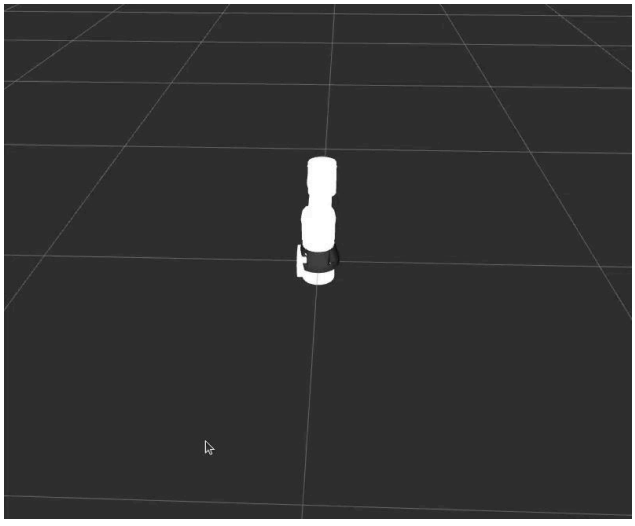


Figure B.14 – RViz depiction of simulated line following. Source [3], slide 27.

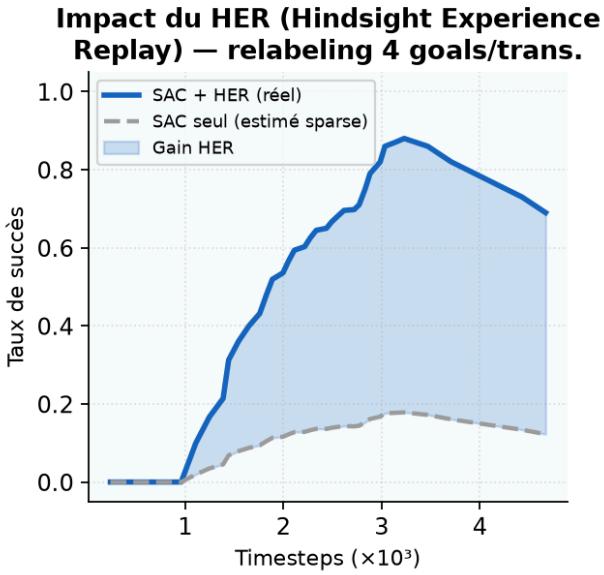


Figure B.15 – Pick-and-place episode horizon and finish. Source [3], slide 30.

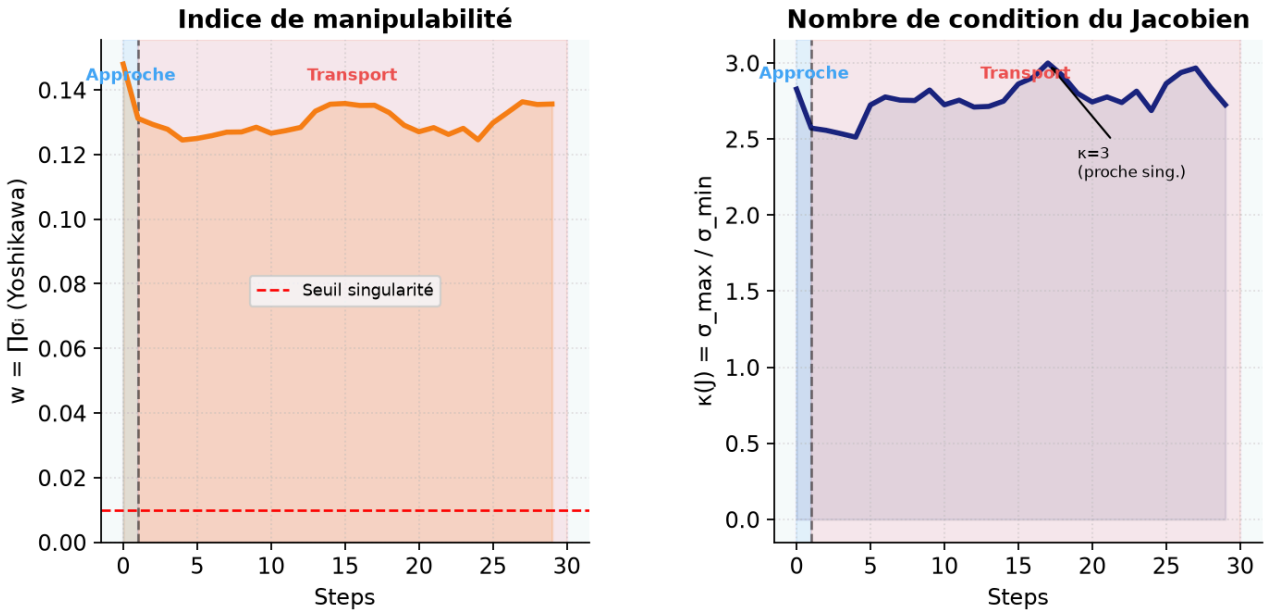


Figure B.16 – Reported pick-and-place auxiliary metrics. Source [3], slide 31.

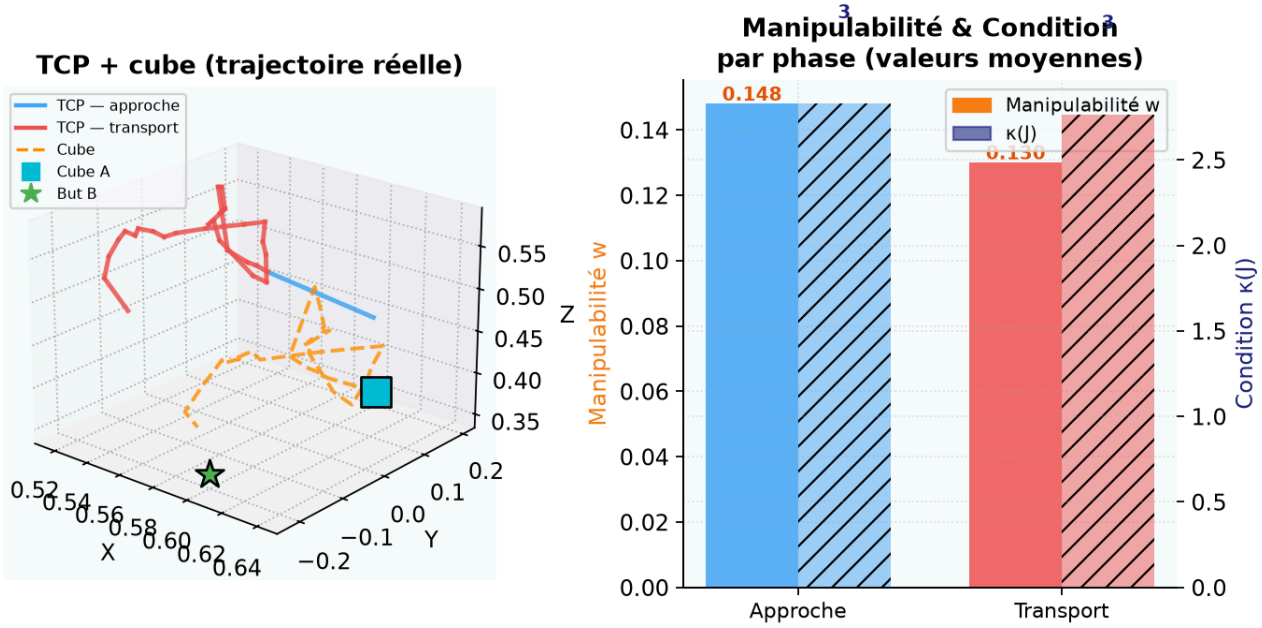


Figure B.17 – Pick-and-place pose and comparison figure. Source [3], slide 31.

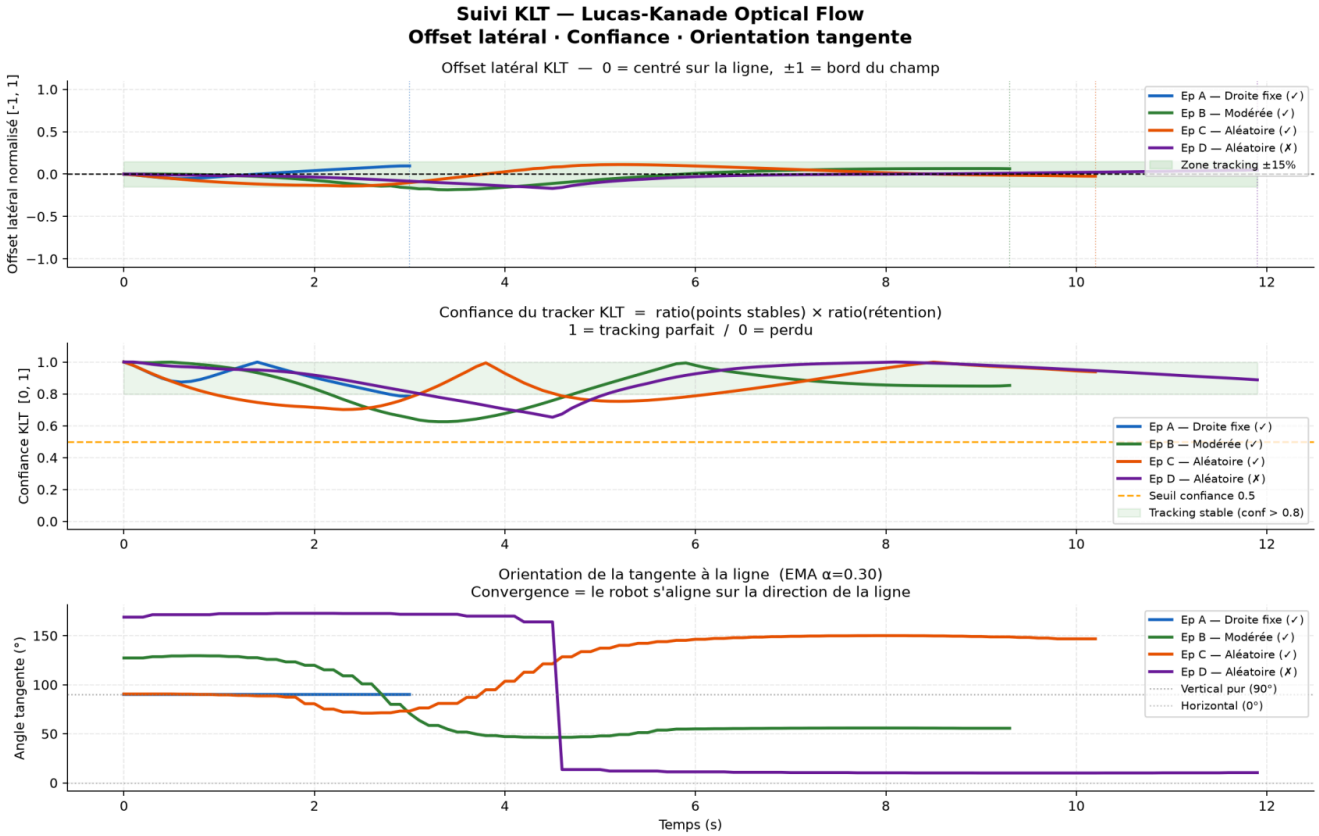


Figure B.18 – KLT quantities in the extended presentation. Source [3], slide 35.

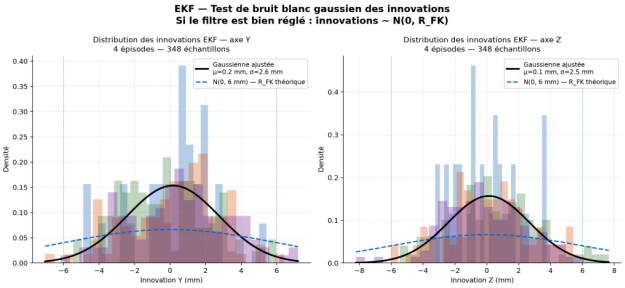


Figure B.19 – FK noise histograms. Source [3], slide 36.

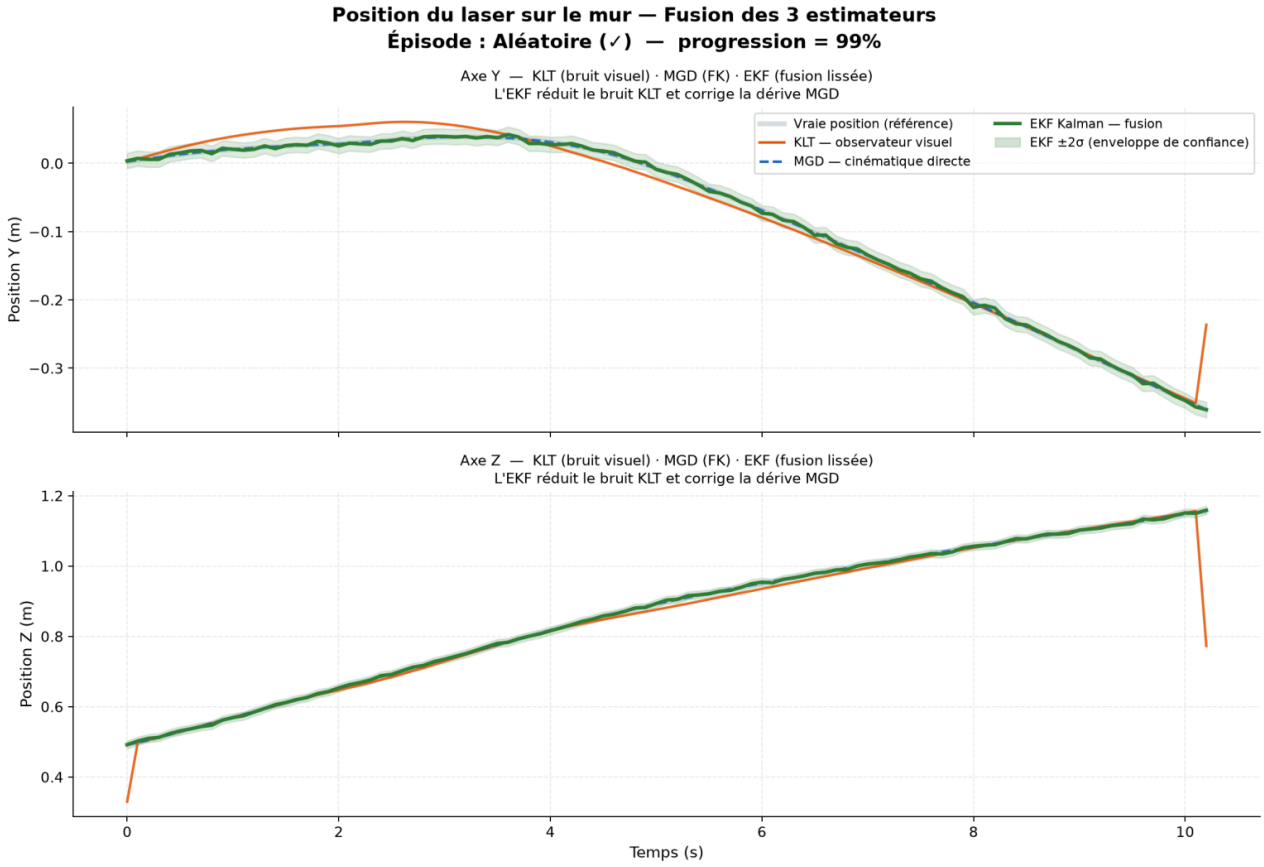


Figure B.20 – Wall-coordinate time records. Source [3], slide 36.

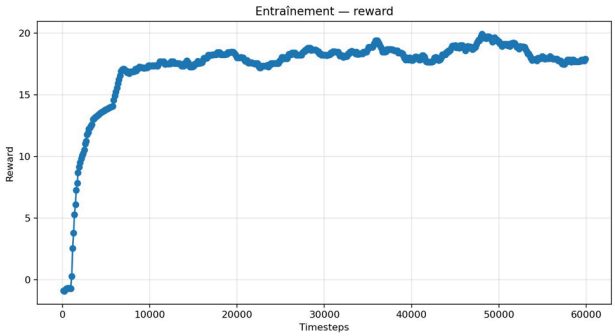


Figure B.21 – Presentation training curve, first panel. Source [3], slide 37.

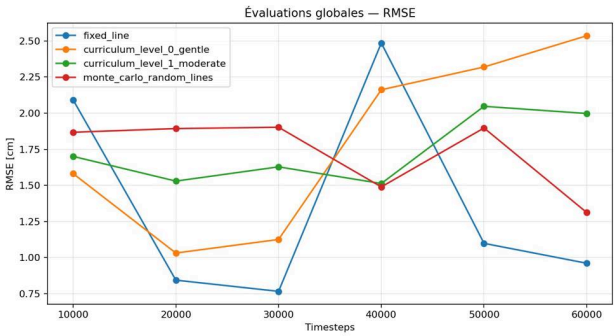


Figure B.22 – Presentation training curve, middle panel. Source [3], slide 37.

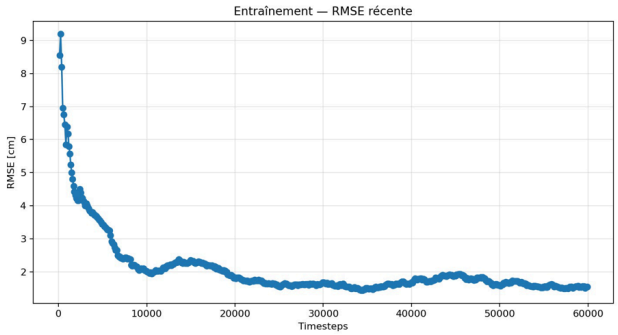


Figure B.23 – Presentation training curve, third panel. Source [3], slide 37.

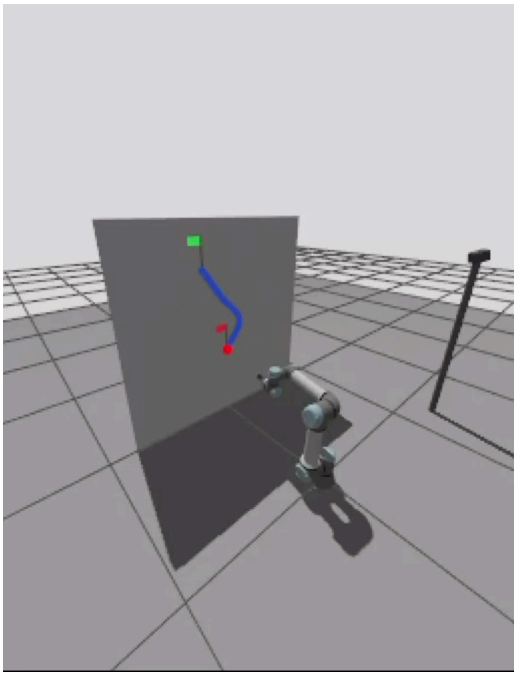


Figure B.24 – Wall-following Gazebo visualisation. Source [3], slide 39.

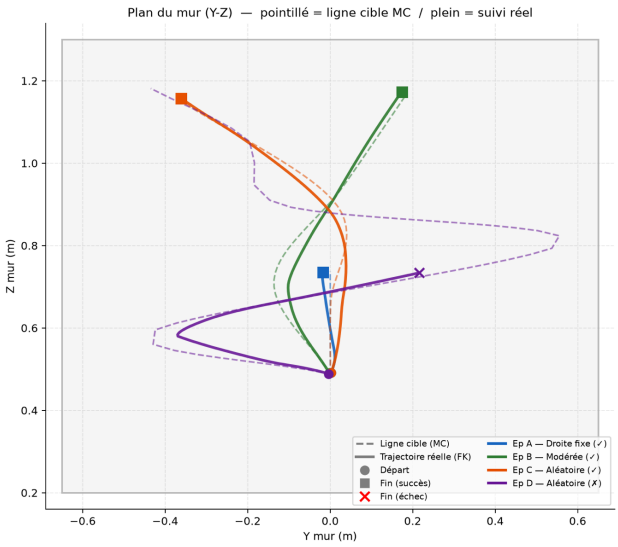


Figure B.25 – Reported complete-model wall paths. Source [3], slide 39.

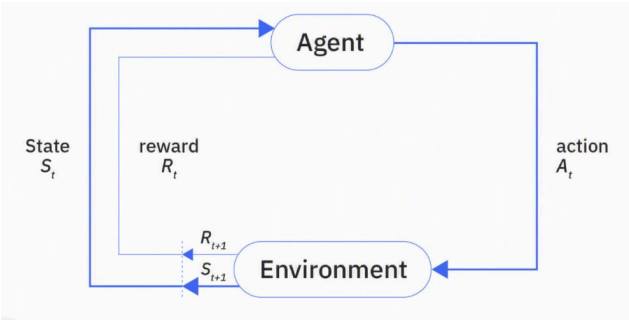


Figure B.26 – Interaction loop shown in the original slides. Source [2], slide 6.

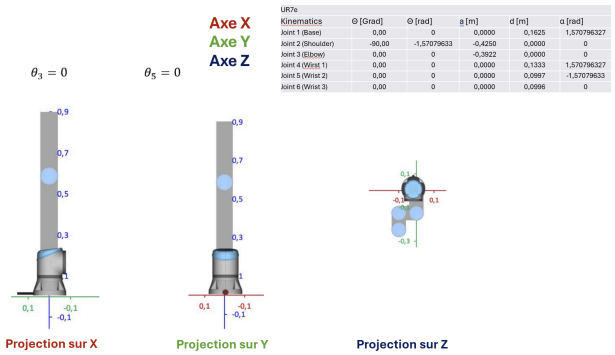


Figure B.27 – Early nominal kinematic sketch, base and arms. Source [1].

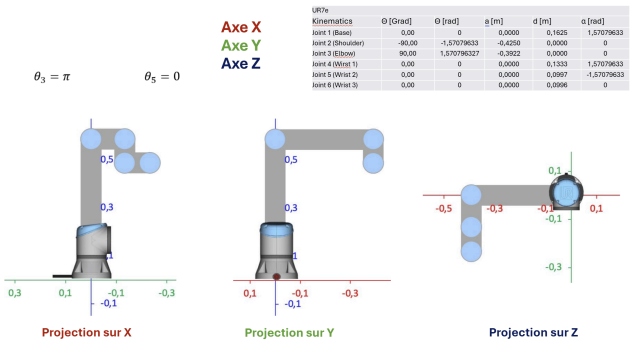


Figure B.28 – Early nominal kinematic sketch, alternate pose. Source [1].

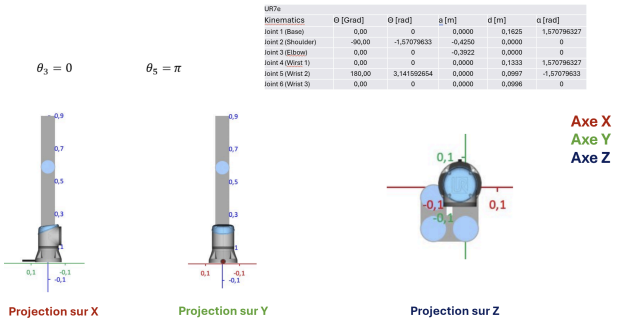


Figure B.29 – Early wrist and arm geometry sketch. Source [1].

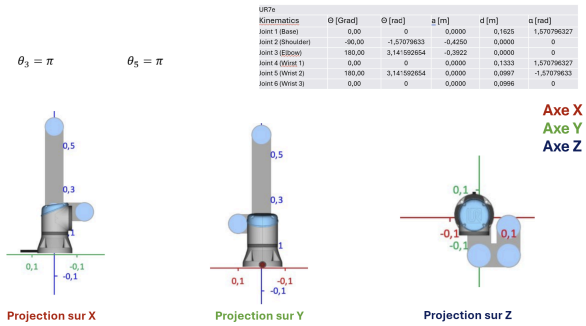


Figure B.30 – Early kinematic projection sketch. Source [1].

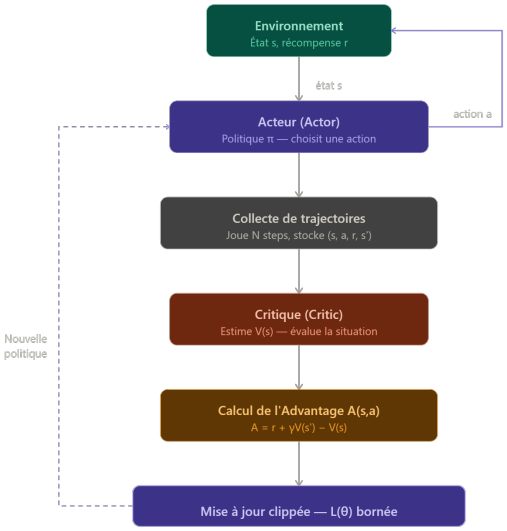


Figure B.31 – Original PPO illustration from the March report. Source [1].

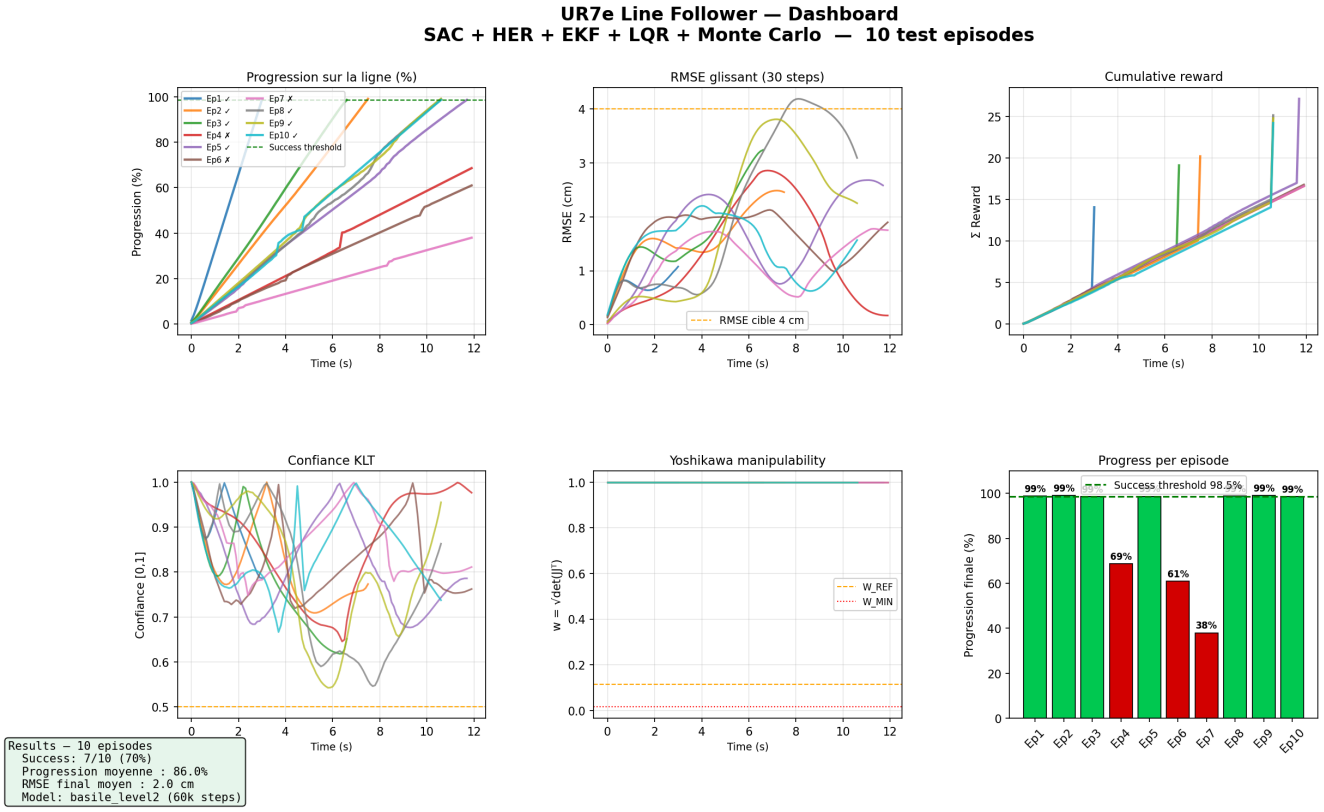


Figure B.32 – Complete-model offline evaluation dashboard. Source [6].

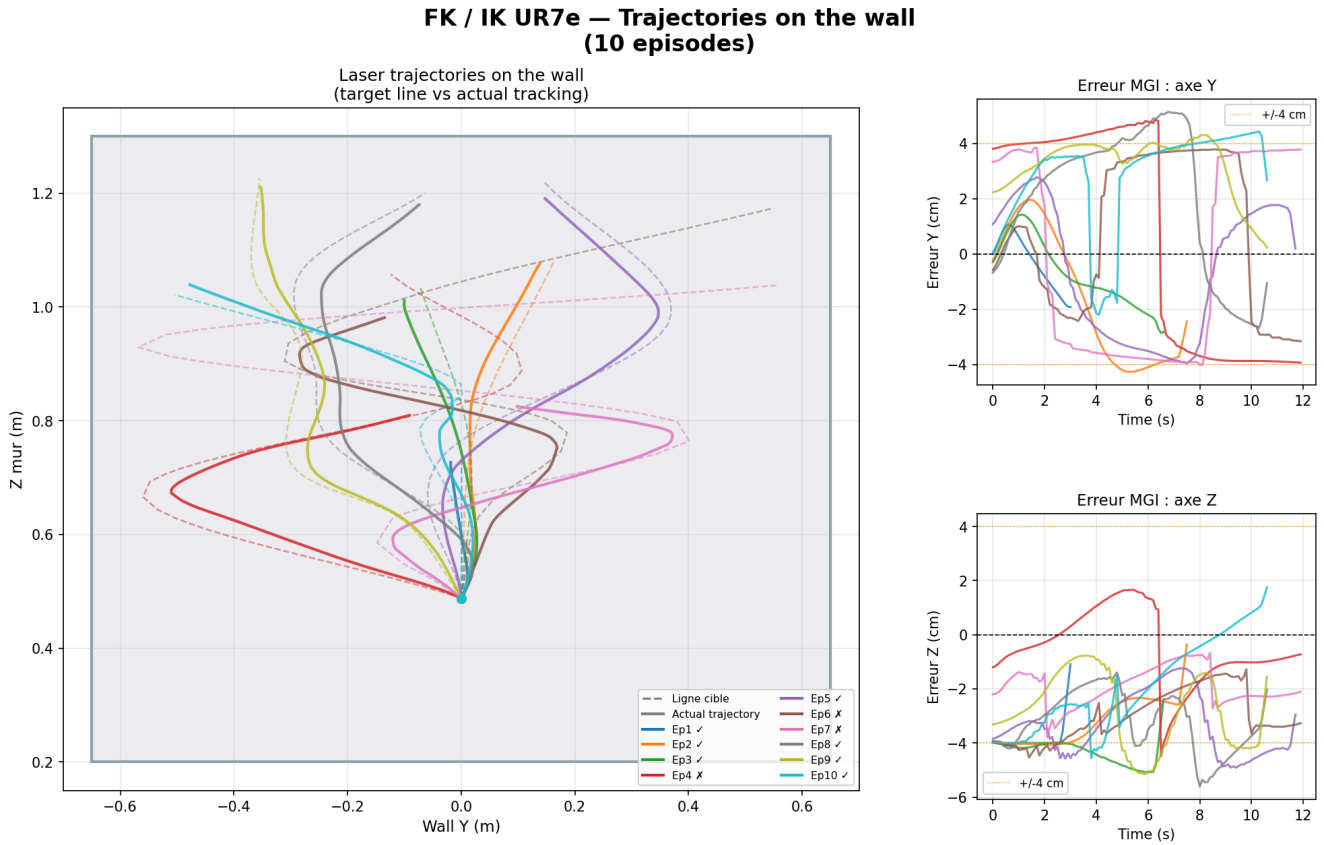


Figure B.33 – Offline MGD/MGI and wall trajectories. Source [6].

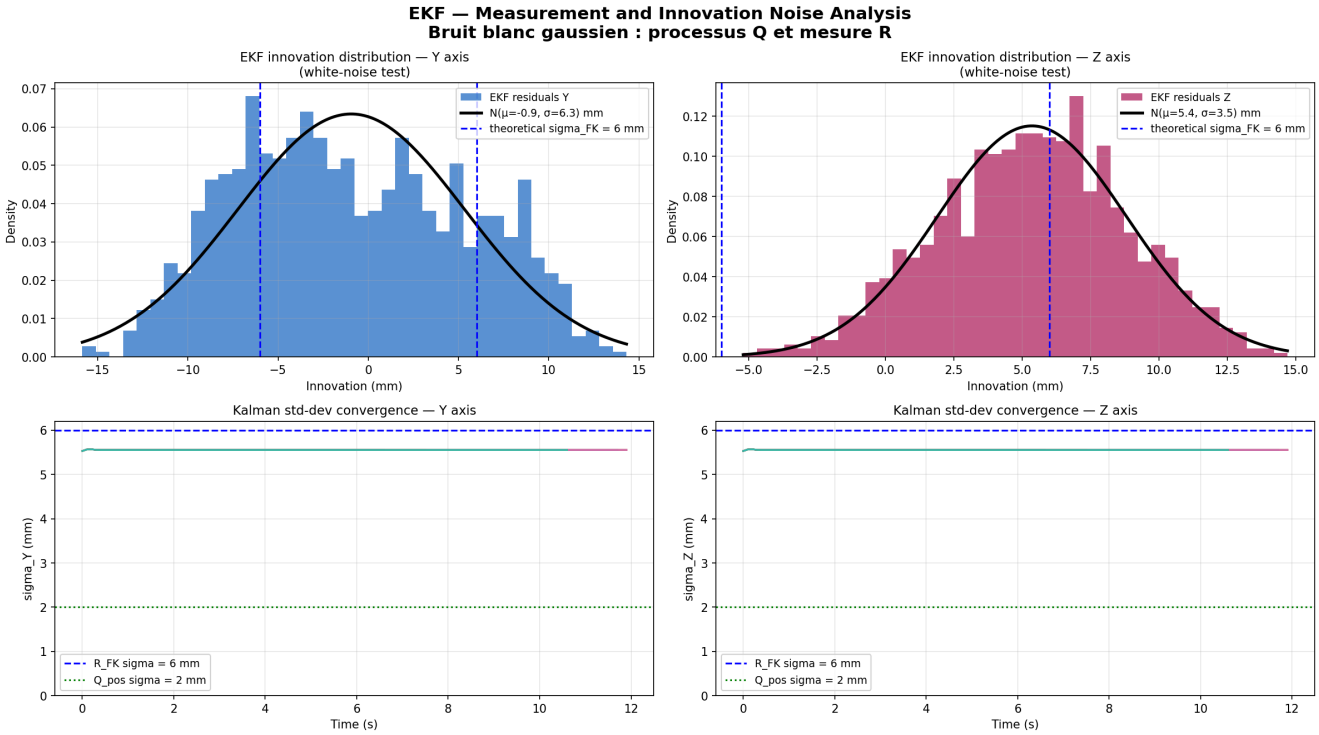


Figure B.34 – Offline noise and observer detail. Source [6].

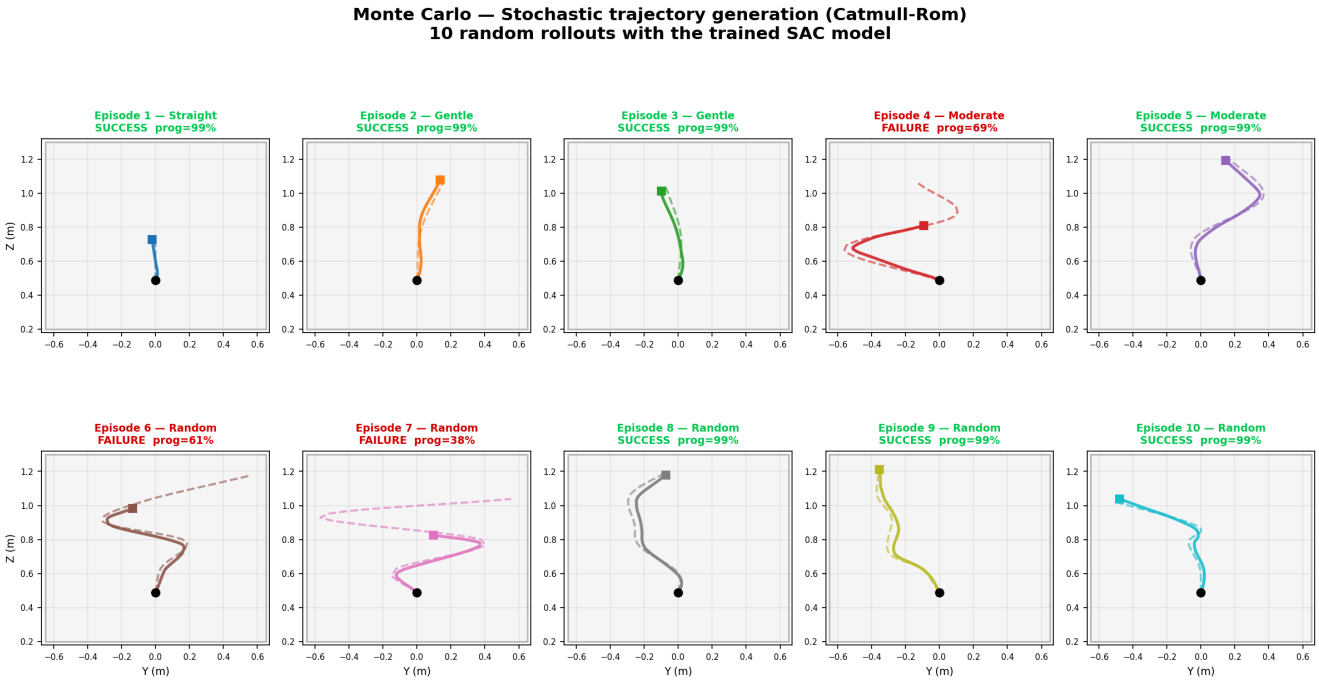


Figure B.35 – Offline example Monte Carlo paths. Source [6].

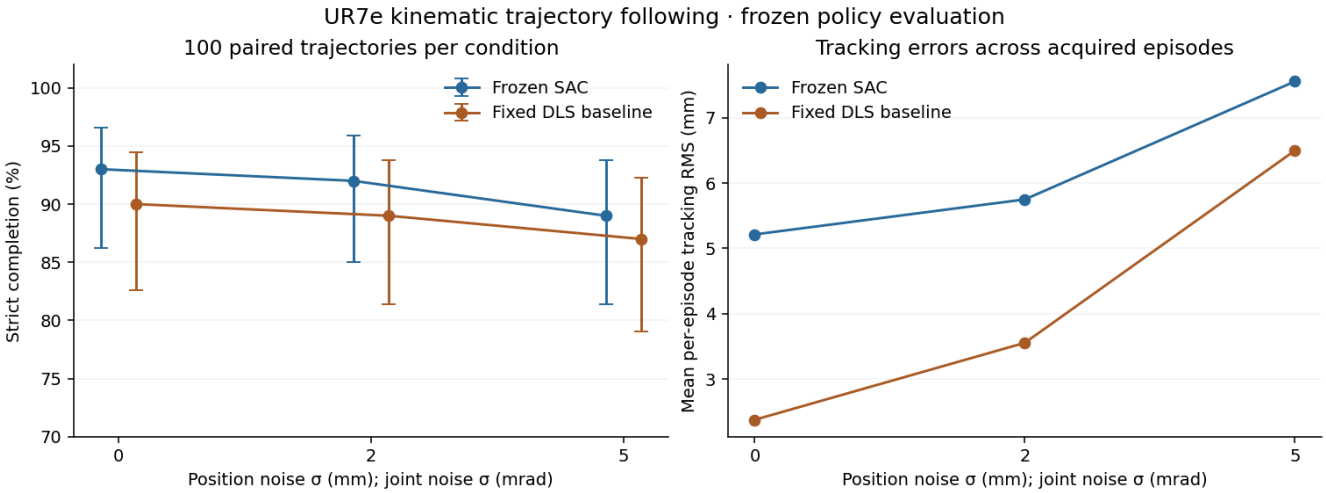


Figure B.36 – Published SAC/DLS benchmark comparison. Source [8].

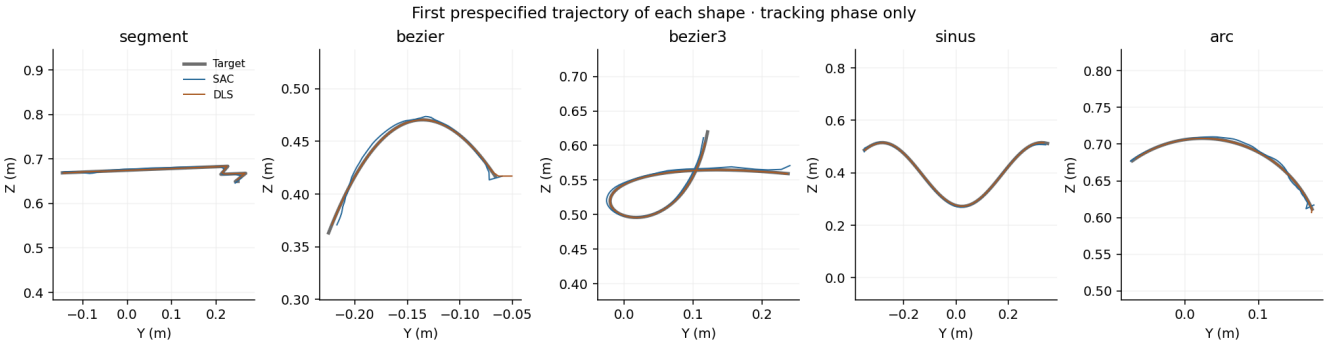


Figure B.37 – Published example paths for frozen SAC and DLS. Source [8].

Appendix C. Data, software and reproduction

The attached *UR7e_reproduction.zip* contains the analysis scripts, the simulated benchmark and training episode records, the replay results, the source and benchmark manifest, and a copy of the MIT-licensed portable actor and its required implementation files. Its checksums and third-party robot-description notices accompany the data. Original software repositories and the model, log, notebooks and live demonstrator are linked at their verified revisions in the references. The data statement does not imply that the March and June project materials contain all underlying logs.

Table C.1 – Data and code files reproduced in the manuscript.

Artifact	Unit of analysis	Use
Frozen benchmark	600 rows / 100 paths	Paired comparison, label audit and replay
Monitor log	520 episodes, one run	Training summary and window trend
Offline wall replay	10 cases, 1 actor	Repeat of complete-model geometry surrogate
UR5e notebook	One synthetic sequence	Reward threshold and hindsight labels
Additional Python tests	Nominal model or seeded trial	FK checks, optics, homography and observer
Source manuscript assets	Independent source panels	Archival figures with named provenance

To rerun the paired benchmark, use Python with NumPy, Gymnasium and IKPy and the archived *release* directory. Extract the attachment using a reader with PDF-attachment support. The data-only analysis and geometry/visual simulations additionally use SciPy, pandas, Matplotlib and OpenCV. The independent wall replay uses CPU PyTorch with tensor-only loading, the archived training package and Stable-Baselines3 as an import dependency. Runtime versions and command lines are recorded in the archive. No hardware connection is part of any of these commands.

```
python replay_ur7e.py --release release --out inputs/replayed.csv
python analysis_ur7e.py --inputs inputs --out results
```

The completed-episode sequence cannot identify the exact sampled line on every training iteration. The 180° and 360° curves and the fixed-obstacle results lack released row-level predecessors for the archived figures. Supporting scene files for the separate PyBullet branch are required for a full rerun. The UR5e tutorial reconstructs the reward and relabelling correspondence, while the full physics manipulation, image-based ROS stack and real-robot validation are separate experimental tasks.

Theory supplement

The theory boxes inserted after Sections 2, 3, 4, 5, 6, 9 and 12 derive the models used to read the results and add model-based checks. Their numerical statements are produced by the separately attached *theory_ur7e.py*, which writes *theory_results.json* and the three box figures from the archived *analysis_results.json*. It uses NumPy, SciPy and Matplotlib only, repeats the observer test of Section 5.3 with the same random stream, and has no hardware interface. Closed forms such as (B2.4), (B3.15) and (B5.13) are checked against direct computation.

```
python theory_ur7e.py --out theory \
    --results results/analysis_results.json
```