Applied Artificial Intelligence and Computing Systems
https://www.scigate.org
Science Gate Academic
Vol. 5, No. 6, 2026

Article

Constraint Aware Handover Scheduling Under Intermittent Occlusion

Maren Veylen*, Oren Talwick, and Elira Nanswick

Department of Mechanical Engineering, University of Saskatchewan, Saskatoon, Canada

*Corresponding author: marenveylen426@outlook.com

Abstract: ***Intermittent visibility creates a scheduling dilemma in human–robot handover: an early approach can intersect an uncertain reach, while a universal waiting buffer can accumulate queueing delay. This study examines that dilemma with a fully specified synthetic event simulator rather than robot experiments. A finite candidate scheduler chooses an approach time and one of two abstract routes using an observation-age uncertainty margin and explicit lateness cost. Four transparent policies are compared on 100 simulated runs containing 48 synthetic requests each, with common latent perturbations across policies. In these simulated data, the occlusion-aware policy reduces precontact threshold violations from 21.54% for mean-only scheduling to 2.92%, while increasing mean request-to-completion time from 1.899 to 2.983 s. An unconditional detour and waiting buffer reaches 0.77% simulated violations but incurs 8.206 s mean flow time in the deliberately congested finite queue. Sensitivity analyses expose uncertainty miscalibration, occlusion-duration effects, and candidate-budget tradeoffs. A constructed hidden-bias trajectory violates the threshold despite satisfying the scheduler's forecast margin, establishing the absence of a physical safety guarantee. The contribution is an auditable systems argument about timing, information, and queueing under stated synthetic assumptions; it supplies no evidence of real-world performance or human acceptability.***

Keywords: human–robot handover; temporal scheduling; intermittent occlusion; uncertainty margins; synthetic simulation; queueing tradeoffs

1 Introduction

Object handover couples an exchange of control with an agreement about timing and location. A robot can arrive at a geometrically reachable configuration and still produce an awkward interaction if its movement conflicts with the receiver's reach. Conversely, a motion that appears conservative when considered in isolation can make a sequence of requests inefficient because every additional pause occupies the same shared resource. This paper develops a deliberately synthetic systems study of that tension. It asks how a finite scheduling rule should trade delayed approach, route clearance, and accumulating queueing time when visibility of the receiving hand is intermittently unavailable. The purpose is to make the assumptions and consequences inspectable, not to establish the performance of a physical robot.

The handover literature distinguishes processes before contact from the physical transfer itself. That distinction is essential here. The review by Ortenzi and colleagues organizes handovers around prehandover coordination and physical exchange, and emphasizes the need for multiple complementary evaluation measures [1]. We retain only an abstract precontact clearance process and a sequence of timed phases. The transfer phase has a prescribed duration; grasp stability, grip force, slip, object

ownership, and deliberate taking intent are not simulated. A threshold violation in this paper therefore means that a scalar synthetic precontact clearance surrogate falls below its chosen numerical threshold. It does not mean that a measured collision occurred or that an actual person was endangered.

Intermittent occlusion makes the distinction between a good mean prediction and a useful decision especially clear. When the nominal reach remains outside a threshold but its uncertainty grows, a scheduler can proceed, wait for a predicted visibility recovery, or use a route with more nominal clearance. Each action has a different effect on both the current request and subsequent requests. Waiting can move an approach beyond an uncertain interval, but waiting also pushes the robot's next availability time forward. A policy that always waits can consequently appear excellent on a clearance metric while imposing a large completion burden on the team. An informative comparison must expose both outcomes with the same underlying synthetic requests.

We examine four elementary policies rather than presenting a trained policy or an elaborate optimizer. The earliest policy immediately takes the direct route. The fixed-buffer policy always adds a waiting interval and a detour. The mean-only policy searches candidate schedules using nominal clearance. The occlusion-aware policy conducts the same search with an uncertainty margin that grows during the designed occlusion interval. These choices isolate different sources of conservatism. They also make it possible to reproduce every result with a short deterministic program, without simulator assets, network services, learned weights, or a proprietary robot interface.

The resulting contribution is a bounded methodological demonstration. First, it specifies the synthetic event process and the exact candidate-selection objective, including their deliberately favorable information assumptions. Second, it compares threshold violations, request-to-completion time, lateness, detour share, and computational work across common random scenarios. Third, it constructs cases in which a forecast margin is misleading and examines sensitivity to assumed uncertainty, occlusion duration, and search resolution. Every numerical result and every data figure or table is simulated. None of its findings constitutes empirical evidence for deployment, certification, or human preference.

2 Related Research and Scope

2.1 Coordination before and during transfer

Strabala and colleagues describe handover coordination as involving both physical movement and agreement about what, when, and where an exchange will occur [2]. This perspective motivates the present separation between scheduling and transfer mechanics. Choosing an approach start time is only one part of coordination. A receiver might decline an object, move attention elsewhere, or change the intended handover location. Our synthetic requests encode none of those decisions. Each request is assumed to represent an accepted exchange with a prescribed service sequence. This simplification allows timing effects to be isolated, while making the gap to an interactive system explicit.

Handover configuration also has dimensions that cannot be reduced to Euclidean clearance. Cakmak and colleagues studied preferences for robot and object configurations and reported differences between configurations derived from human examples and those generated with a kinematic model [3]. Dragan and colleagues distinguished legibility from predictability in robot motion [4]. These studies caution against interpreting a detour as inherently better simply because it raises a numerical clearance. A longer or differently shaped approach could communicate intent poorly, impose an uncomfortable reach, or conflict with task expectations. The current objective contains no model of those responses, so it reports detour share as a resource choice rather than a preference score.

Learning approaches address additional aspects of adaptation. Kupcsik and colleagues formulated dynamic robot-to-human handover using contextual policy search and human feedback [5]. More recently, the 2026 preprint by Marra and colleagues combines temporal tactile information and compliance for intent-aware bimanual transfer [6]. These are relevant because visual uncertainty does

not disappear when the robot reaches the transfer pose. Tactile history and force regulation concern the exchange itself. Our simulator deliberately stops short of those mechanisms: it does not train a model, estimate taking intent, or claim to reproduce the experiments in either source.

2.2 Temporal plans and uncertainty-aware motion

Task-level coordination offers a complementary view of the problem. Chaski schedules robot actions in shared plans and adapts execution to a human teammate [7]. Human–robot cross-training explicitly considers the shared plan learned through role switching [8]. The present simulator uses a much smaller temporal structure, with requests served in release order and no reassignment of roles. Nevertheless, these works motivate evaluating idle or waiting time together with local motion properties. A decision can improve one interaction while degrading the larger process that contains it.

Temporal constraint networks provide a general language for expressing allowable differences between event times [9]. Our release, start, and completion constraints are a small special case. We do not claim to solve general temporal planning, nor do we permit request reordering. The finite search decides only the start delay and route for the next request. This makes the consequences of the uncertainty rule easier to interpret, although it excludes the potentially important alternative of serving another task while a handover is temporarily unattractive.

Confidence-aware human prediction research connects uncertainty estimates to robot planning rather than treating a forecast mean as sufficient [10]. Chance-constrained path planning similarly makes explicit the relationship between probabilistic uncertainty and constraint satisfaction [11]. Earlier probabilistic conflict-detection work distinguishes instantaneous conflict measures from encounter-level questions [12]. We draw a narrow methodological lesson from these sources: a clearance margin must be interpreted together with the model, the event definition, and the horizon over which it is checked. Our finite scalar screen lacks the assumptions needed to inherit their guarantees.

2.3 Relation to control guarantees and simulation validation

Control barrier functions can express forward-invariance conditions through restrictions on admissible control inputs [13]. Discrete-time barrier formulations have also been incorporated into model predictive control [14]. Our scheduler is not a barrier controller. It has no joint dynamics, no actuator constraints, no control-input optimization, and no proof that a chosen approach remains in an invariant set. Calling its inequality a constraint is a statement about candidate admission in the synthetic model. It should not be read as an invariance result for robot motion.

A similar separation applies to learning-based predictive control and tracking guarantees. Koller and colleagues use reliable uncertainty descriptions and terminal conditions to reason about safe exploration [15]. FaSTrack explicitly separates simplified planning models from higher-dimensional tracking dynamics and provides a tracking-error bound [16]. We have neither a learned dynamics model nor a tracking subsystem. Candidate paths are abstract alternatives with prescribed time and clearance changes. Sampling-based planning theory also concerns a search space and asymptotic properties that do not apply to our two-route enumeration [17]. These distinctions prevent a simple scheduling illustration from borrowing stronger language than its construction supports.

Finally, simulation verification and real-system validation are different activities. Sargent's account of simulation-model verification and validation emphasizes conceptual validity, implementation correctness, operational validity, and data validity as distinct questions [18]. We can inspect the event logic, reproduce seeded outcomes, and identify model counterexamples. We cannot validate the numerical inputs against an operational cell because no such measurements were collected. The study therefore treats sensitivity analysis as an examination of internal behavior, not as a substitute for external evidence.

3 Synthetic Scheduling Model

3.1 Requests, service phases, and available information

Each simulated run contains forty-eight requests served by one abstract robot in release order. The release sequence begins at zero; subsequent interarrival intervals are independently drawn from a uniform distribution between 2.35 and 3.15 s. Each request receives a due time between 2.8 and 4.0 s after its release, a difficulty variable uniformly distributed between zero and one, an occlusion onset between 0.1 and 0.8 s after release, and an occlusion duration between 0.65 and 1.8 s. These distributions are chosen design inputs. They do not summarize factory observations, human reaction times, or a robot manufacturer's performance specifications.

The robot's availability is determined by the preceding request's completion. A start cannot precede either release or availability. Writing the release time as r, approach start as s, and preceding completion as c gives the event constraint below. The first request uses a preceding completion of zero. This recursion creates queueing even though requests cannot overlap in service. It also means that a policy's earlier decisions change the candidate times encountered later in a run. Policies consequently share requests and latent disturbances, but need not encounter them at the same elapsed time.

$$s_i \geq max\left(r_i, c_{i-1}\right) \tag{1}$$

A binary route variable selects a direct approach or a detour. The detour increases approach duration by 0.35 s and raises the nominal clearance surrogate by 0.18 m. Difficulty increases approach duration and transfer duration, while retraction remains fixed at 0.45 s. The operation completes after all three phases. This service decomposition is intentionally small enough that queueing effects can be traced without an underlying rigid-body engine. No acceleration, torque, contact force, or end-effector orientation is represented by these time terms.

$$a_i(q) = 0.78 + 0.28\,u_i + 0.35\,q \tag{2}$$

$$c_i = s_i + a_i(q) + 0.40 + 0.10\,u_i + 0.45 \tag{3}$$

The scheduler is given the designed nominal clearance function and the complete future occlusion interval for the current request. That is a strong information assumption. It resembles a cell with a known moving occluder schedule more closely than a camera system trying to predict arbitrary human occlusion. A real system would need to infer the interval, revise it when observations disagree, and handle the possibility that visibility never recovers when expected. The present results should be interpreted conditional on this explicit information advantage. All search-based policies receive the same interval and nominal model; the mean-only policy simply does not use the uncertainty scale when screening candidates.

Table 1. Synthetic design inputs used by the event simulator; none are measured robot or human parameters.

Quantity	Synthetic value or distribution	Role
Requests per run	48	Finite service queue
Independent seeded runs	100	Main comparison units
Interarrival interval	Uniform 2.35–3.15 s	Request load
Occlusion onset	Uniform 0.10–0.80 s	Visibility timing
Occlusion duration	Uniform 0.65–1.80 s	Uncertainty growth interval

Quantity	Synthetic value or distribution	Role
Precontact threshold	0.18 m	Synthetic violation definition
Candidate delay grid	0.00–4.50 s in 0.15 s steps	Finite search budget

3.2 Nominal clearance and latent perturbations

The nominal clearance surrogate varies smoothly with time from request release. Its baseline is 0.36 m, with a sinusoidal component of amplitude 0.075 m and a request-specific phase. A Gaussian-shaped pulse centered halfway through the occlusion interval reduces nominal clearance by up to 0.15 m. This pulse is not inferred from an image. It is a transparent device for creating a time interval during which the direct route becomes less attractive. The same nominal function is used for candidate evaluation and for generating the realized surrogate before disturbance terms are applied.

$$g_i(t) = \exp\left[-\frac{(t - r_i - o_i - 0.5\,d_i)^2}{2(0.43)^2}\right] \tag{4}$$

$$\mu_i(t, q) = 0.36 + 0.075 \sin(\phi_i + 1.1(t - r_i)) - 0.15\, g_i(t) + 0.18\, q \tag{5}$$

An observation-age scale increases while the designed occlusion is active. Let A denote elapsed time since occlusion onset, clipped between zero and the designed duration, and let I indicate that the sample lies inside the interval. The scale contains a baseline of 0.035 m, a growth term of 0.105 m per second of occlusion age, and a small approach-horizon term. At the planned visibility recovery, the growth component resets immediately. This discontinuity is intentional and visible in the design illustration. It expresses an idealized measurement update, not the behavior of a calibrated estimator.

$$\sigma_i(t, s) = 0.035 + 0.105\, A_i(t)\, I_i(t) + 0.012(t - s) \tag{6}$$

The realized signed-clearance surrogate subtracts a shared Gaussian perturbation multiplied by that baseline scale, a sample-specific residual, and an occasional hidden bias pulse. The shared perturbation is drawn once per request, so the twelve approach samples are correlated. Residuals have standard deviation 0.012 m and are held fixed across policies for the same request. With probability 0.065, a hidden bias amplitude is drawn uniformly between 0.10 and 0.19 m. These residual and bias terms are deliberately absent from the scheduler's uncertainty formula. They create model discrepancy rather than quietly making the generator identical to the decision rule.

$$y_{ij} = \mu_i(t_{ij}, q) - \sigma_i(t_{ij}, s)\, z_i - \epsilon_{ij} - b_i\, g_i(t_{ij}) \tag{7}$$

Negative values of this signed surrogate are mathematically allowed. They represent penetration of an abstract clearance coordinate, not a physically meaningful negative Euclidean distance. The workspace sketch places the timing problem in a geometric setting, but the plotted arm and uncertainty ellipses do not define a forward-kinematics model. We avoid interpreting the numbers as verified distances between robot links and body segments. This distinction is especially important for the constructed failure example, where a large hidden bias drives the surrogate below zero.

3.3 Sampling and the meaning of a threshold violation

Each candidate approach is evaluated at twelve equally spaced normalized positions between 0.06 and 0.94 of its approach duration. Endpoints are excluded to keep the screen separate from intentional transfer contact. The event evaluator uses exactly these same twelve sample locations for the selected approach. A violation is recorded if at least one realized value is below 0.18 m. The metric is therefore a

sampled precontact surrogate event. It does not inspect continuous-time minima between samples, retraction motion, object geometry, the human's torso, or any other robot link.

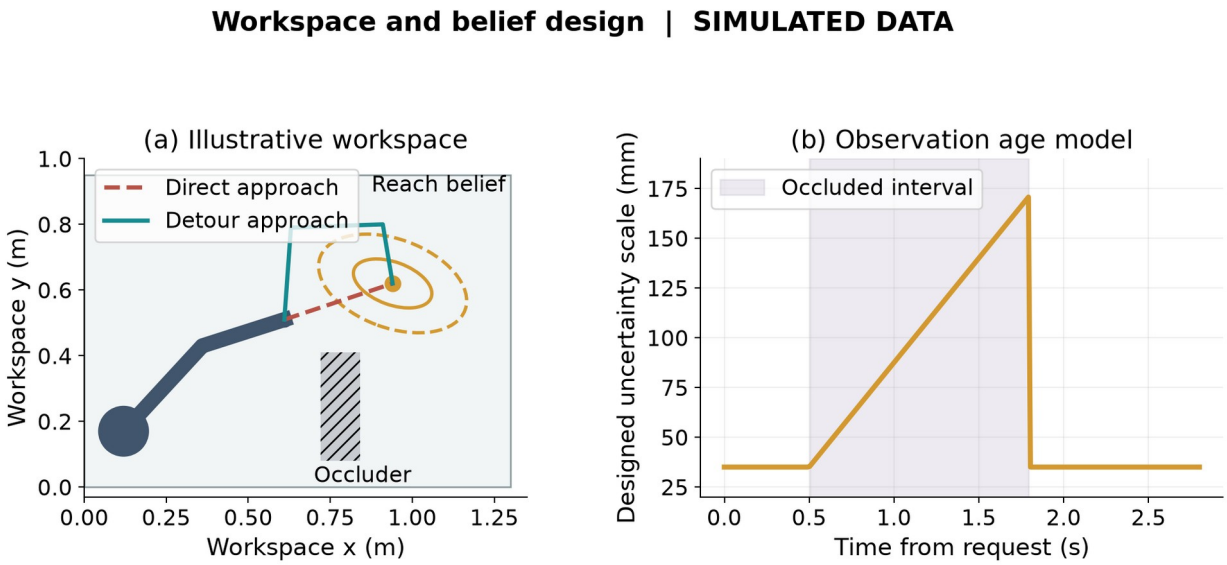


Figure 1. Simulated design illustration of the robot workspace and observation-age uncertainty. Panel (a) is a schematic geometric explanation with two abstract approach routes and uncertainty ellipses; it is not a robot photograph or a kinematic simulation. Panel (b) shows a synthetic uncertainty-growth example with an idealized reset after the occlusion interval.

This choice is restrictive but useful for interpretability. A smaller sample count would miss more structure if the nominal function had rapid variation. A larger count could find additional minima and would require reconsidering the relationship between a pointwise margin and an approach-level event. Because our generator is smooth apart from the visibility reset and because its mathematical form is disclosed, the sampling rule can be inspected directly. Nevertheless, matching the evaluator to the candidate screen does not validate either against a real sensor or a real collision checker. It only ensures that numerical comparisons refer to the same defined event.

The simulator assumes that request-specific disturbances remain associated with normalized approach samples when a policy changes the start time. This is a common-randomness coupling for comparison. It permits a controlled question about how scheduling transforms exposure to the same latent request difficulty, correlated deviation, and residual sequence. Another coupling, such as a global continuous disturbance field indexed only by wall-clock time, could yield different paired outcomes. We therefore report policy means across many synthetic runs and use illustrative paired trajectories to explain mechanisms, without treating a particular paired trajectory as a prediction of what a real person would have done under alternative robot actions.

4 Candidate Admission and Scheduling

4.1 Finite uncertainty screening

For the search-based policies, candidate starts are generated relative to the earliest feasible start. The nominal delay grid contains thirty-one values, including both zero and 4.5 s, and is crossed with two route choices. Each request therefore has sixty-two candidate schedules. A candidate is admitted when its minimum forecast lower margin, evaluated over the twelve approach samples, reaches the synthetic threshold. The uncertainty multiplier k controls the strength of this screen. The main occlusion-aware comparison uses 1.64, while the mean-only policy uses zero.

$$B_i(s,q;k)=\min_j\left[\mu_i(t_{ij},q)-k\,\hat{\sigma}_i(t_{ij},s)\right] \tag{8}$$

The assumed scale is a multiplier of the baseline design scale. It equals the baseline in the main comparison and is deliberately changed in the calibration sensitivity analysis. Importantly, this assumed scale omits the residual and hidden-bias terms present in the generator. Even when its multiplier is one, it is not a complete distribution for realized clearance. The term calibration sensitivity therefore refers to controlled scaling of one forecast component, not to a validated calibration procedure fitted to observations. No estimated coverage curve or held-out prediction set exists in this study.

Among admitted candidates, the scheduler minimizes waiting cost, detour cost, and positive lateness. Waiting is measured from request release, so inherited queueing is included. The detour penalty is fixed. Lateness is computed at operation completion rather than at first approach or first contact. This matters because route choice changes service time even if the robot begins immediately. The weights are selected to create visible tradeoffs; they are not elicited utilities or monetary costs. The objective is dimensionally normalized by these chosen coefficients and has no claim to transfer across workcells.

$$J_i(s,q) = 0.7(s - r_i) + 0.45\,q + 3\,max(c_i - d_i^{due}, 0) \tag{9}$$

The admitted candidate with minimum objective is selected; ties follow deterministic enumeration order. If no candidate is admitted, the code selects the candidate with the largest lower margin and records a recovery flag. This fallback is a way to keep a synthetic run defined, not an acceptable substitute for a deployment-specific stop or withdrawal policy. No recovery is needed in the reported main runs or parameter sweeps. That observation should not be generalized to a larger parameter range. In particular, an occlusion longer than the finite search horizon or a lower baseline clearance could eliminate every candidate.

4.2 Four policies and what they isolate

The earliest policy isolates the cost of ignoring both nominal clearance and uncertainty. Because its direct operation duration remains below the minimum interarrival interval, it starts every request at release in the main synthetic scenario. The absence of queueing under this policy is a consequence of the chosen service and arrival distributions. It is not an empirical claim about a fast robot. The fixed-buffer policy always delays 0.8 s after availability and always selects the detour. It is intentionally easy to audit, but the added delay is paid again for each request, including when visibility is already good.

The mean-only policy searches the full candidate set while checking only nominal clearance. It can therefore use a detour when a direct approach violates the nominal threshold, without paying for uncertainty that it does not represent. The occlusion-aware policy uses the same objective, routes, grid, and nominal function, adding only the uncertainty subtraction. This pair provides the most direct mechanism comparison. Their differences can still propagate through later queue states, so the aggregate effect is not merely the duration added to one isolated request. A schedule is a sequence of coupled availability decisions.

Synthetic finite-candidate handover scheduler

Read the next request and set availability to the later of release and preceding completion.

Enumerate thirty-one delay values and both abstract route choices.

Evaluate nominal clearance and assumed uncertainty at twelve approach samples for every candidate.

Admit candidates whose minimum lower margin reaches the synthetic threshold.

Select the admitted candidate with minimum waiting, detour, and positive-lateness cost.

If none is admitted, select the maximum-margin candidate and record the synthetic recovery flag.

Generate realized surrogate samples with the fixed latent request perturbations, record metrics, and advance availability.

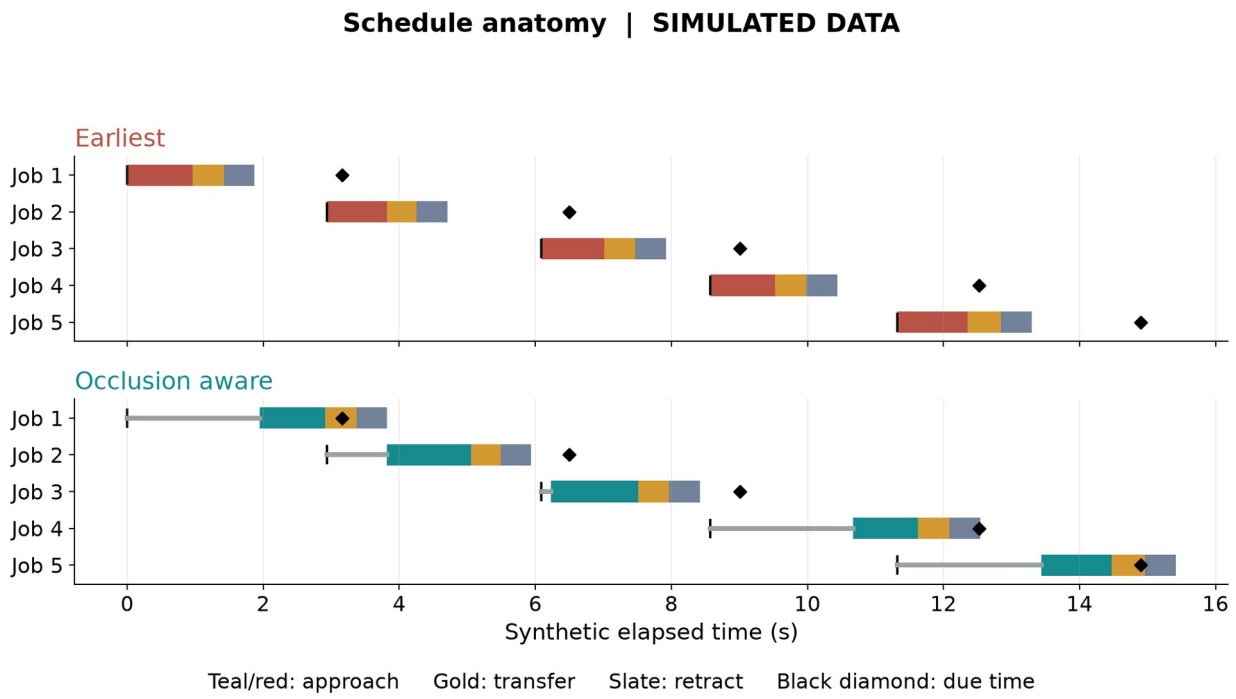


Figure 2. Simulated Gantt schedules for the first five synthetic requests of seed 73219. The earliest and occlusion-aware policies share request release times and due times, while their approach timing and route durations differ. Colored blocks distinguish approach, prescribed transfer, and retraction; thin waiting segments expose the effect on the finite queue.

4.3 Why the margin is not a safety certificate

For one scalar Gaussian perturbation with an exactly known scale, no additional residual, and no hidden bias, an inequality based on k can be related to a one-point tail probability. The expression below states that idealized relationship. Its premises are narrower than the implemented generator. In particular, the selected sample times and routes are decision dependent, the minimum is taken across multiple samples, and the unmodeled residual and bias alter the distribution. We use the expression to identify missing assumptions, not to assign a certified probability to the proposed scheduler.

$$Pr\{y_{ij} < h\} \le 1 - \Phi(k) \tag{10}$$

A union bound would sum pointwise probabilities to upper-bound an approach-level event if suitable pointwise bounds were valid. With twelve points, even the idealized one-sided Gaussian interpretation of 1.64 would produce a loose bound near sixty percent when summed without further structure. Correlation can make that bound conservative, but does not justify replacing it with a five-percent approach guarantee. Moreover, neither the residual nor the hidden bias is included in the assumed scale. The observed simulated violation share of 2.92% therefore cannot be read as confirmation of a nominal probability specification.

$$Pr\{min_j\, y_{ij} < h\} \le \sum_{j=1}^{12} Pr\{y_{ij} < h\} \tag{11}$$

Scenario optimization can provide probabilistic feasibility statements under specified mathematical conditions, including conditions on sampling and optimization structure [19]. Exact feasibility results for randomized uncertain convex programs require corresponding assumptions [20]. Our synthetic search does not invoke those results: its event process contains model discrepancy, its discrete choices are evaluated through a finite heuristic screen, and its reported runs serve comparison rather than a formal scenario-design certificate. A meaningful guarantee would require a different argument that explicitly includes estimation error, continuous motion, and a realizable fallback.

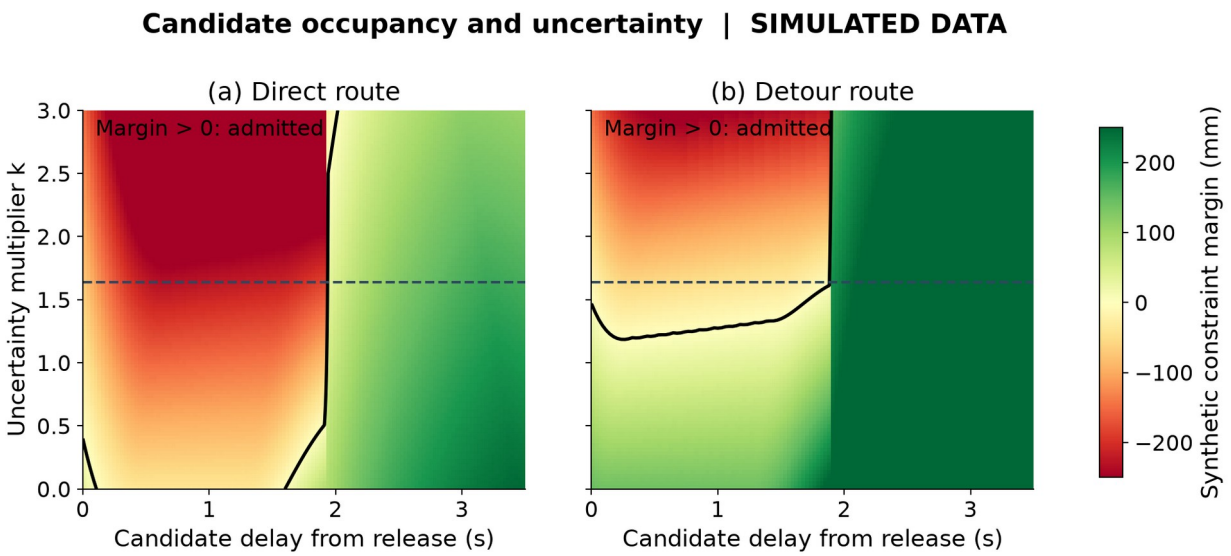


Figure 3. Simulated candidate-admission maps for the first synthetic request. Color shows the minimum forecast margin above the 0.18 m synthetic threshold over candidate delay and uncertainty multiplier. The black contour marks zero margin and the dashed horizontal line marks the main multiplier 1.64. The abrupt boundary reflects the designed visibility reset, not a measured perception transition.

5 Evaluation Protocol and Reproducibility

The main comparison contains one hundred independently seeded synthetic runs, each with forty-eight requests. The base seed is 73219 and subsequent run seeds increment by one. Every policy receives the same generated requests for a given seed. Latent Gaussian deviations, sample residuals, hidden-bias indicators, difficulty, due times, and nominal phases are held fixed across policies. The simulation is deterministic once the seed and design parameters are fixed. The supplied program regenerates all data graphics and a machine-readable metrics file without downloading a dataset or executing a trained model.

We summarize request-to-completion time, waiting time, positive lateness, threshold-violation share, detour share, and candidate-sample evaluations. The first three are measured in synthetic seconds. A request with positive lateness is also counted in a tardy fraction, because a small mean lateness can coexist with many requests that narrowly miss their due times. The violation indicator below uses the minimum of the twelve realized signed-clearance samples. Its denominator is all requests, including those delayed by inherited queueing. No request is removed after an unfavorable outcome.

$$V_i = 1\{\min_j y_{ij} < 0.18\} \tag{12}$$

$$F_i = c_i - r_i \tag{13}$$

Run-level means are the principal comparison units. Requests within a run share a queue state and should not be treated as independent experimental participants. We show distributions of the one hundred run means and compute a paired bootstrap interval for the difference in mean flow time between occlusion-aware and mean-only policies. The bootstrap resamples complete paired runs three thousand times with a separate fixed seed. It describes variability under the synthetic generator. It does not quantify uncertainty about transferring the generator to a real cell or about the correctness of its chosen input distributions.

The uncertainty-multiplier and scale-sensitivity sweeps use forty runs per setting, paired across settings through the same first forty seeds. The occlusion-duration sweep uses thirty runs per setting. Candidate-budget sensitivity uses twenty runs per grid spacing. These smaller sweeps are labeled separately from the main comparison because their estimates need not match the one-hundred-run means

exactly. Reusing seeds reduces random differences when comparing neighboring settings. It does not make the settings independent replications or remove the modeler's influence over the scenario design.

Computational burden is expressed as candidate-sample evaluations rather than an execution-time benchmark. A search with sixty-two candidates and twelve samples requires 744 such evaluations per request. Both fixed policies evaluate twelve realized approach points but do not conduct the same candidate search. This operation count is portable enough to audit, whereas wall-clock timings would depend on vectorization, plotting imports, processor behavior, and the local runtime. The study makes no real-time execution claim. In a physical architecture, sensing, collision checking, communication, and controller synchronization could dominate the small arithmetic workload reported here.

6 Simulated Results

6.1 Main policy comparison

The main simulated results expose a broad separation between fast nominal service and lower threshold exposure. The earliest policy has mean synthetic flow time 1.818 s and records 1,609 violations among 4,800 requests, or 33.52%. The mean-only policy increases flow time to 1.899 s while lowering the violation share to 21.54%. Its nominal screening is therefore helpful under this generator, but many realized surrogate values still fall below the threshold. This is expected when the generator contains correlated deviation and hidden bias that are not removed by a positive nominal margin.

The occlusion-aware policy records 140 simulated violations, corresponding to 2.92%, with mean flow time 2.983 s. Relative to mean-only scheduling, the mean paired increase in flow time is 1.084 s. The paired run bootstrap interval is 1.052 to 1.114 s. These are internal simulation comparisons: the lower violation share and larger flow time are both consequences of the stated timing and uncertainty model. The result does not establish that the same multiplier, wait horizon, or clearance threshold would be appropriate for a particular robot or person.

Table 2. Main simulated outcomes over 100 runs and 4,800 synthetic requests per policy. Violation means sampled surrogate threshold violation, not observed physical collision.

Synthetic policy	Flow time (s)	Wait (s)	Violation (%)	Detour (%)	Mean lateness (s)
Earliest	1.818	0.000	33.52	0.00	0.000
Fixed buffer	8.206	6.038	0.77	100.00	4.825
Mean only	1.899	0.001	21.54	22.81	0.000
Occlusion aware	2.983	1.010	2.92	44.33	0.148

The fixed-buffer policy produces the lowest violation share in this four-policy comparison, 0.77%, but its mean flow time is 8.206 s and mean positive lateness is 4.825 s. Its mean waiting time exceeds the explicit 0.8 s buffer because preceding waits propagate through the finite queue. The design intentionally places its mean per-request occupation above the mean interarrival interval. It is therefore a finite-horizon congestion example, not a neutral representation of all fixed-buffer methods. Releasing requests more slowly, allowing the buffer to overlap other work, or applying the delay only during occlusion would change the result substantially.

The comparison also illustrates why a single ranking would be misleading. Fixed buffering has a lower simulated violation share than the main occlusion-aware setting, while earliest service has lower flow time. The scheduling choice depends on which outcomes are constrained and what operating regime is assumed. The occlusion-aware policy occupies an intermediate region by adapting time and route to its forecast. Its advantage over unconditional buffering in flow time should not obscure that it

still incurs violations and lateness. The results describe a tradeoff surface, not a universally dominant controller.

6.2 Distributional variation and accumulated delay

Run-level variation is visible even though every run uses the same distribution families and request count. The standard deviation of the mean flow time is 0.168 s for the occlusion-aware policy and 1.096 s for fixed buffering. The latter policy's tenth, fiftieth, and ninetieth percentiles of run mean flow time are 6.679, 8.285, and 9.464 s. These values reflect finite queue accumulation under different difficulty and arrival sequences. A slow first portion of a run can push later requests into a different availability regime, while a favorable sequence can partially offset the imposed delays.

For the occlusion-aware policy, the corresponding flow-time percentiles are 2.768, 2.966, and 3.206 s. Its standard deviation of run-level violation share is 2.39 percentage points. Some runs therefore exhibit substantially more violations than the aggregate share alone suggests. The distributions are shown rather than hidden behind one rounded mean. This is particularly relevant for a request-level event with correlated samples: a small number of unfavorable latent deviations can produce an uneven distribution across finite runs without implying any failure of deterministic reproducibility.

The tardy fraction adds another view. Although the occlusion-aware policy's mean positive lateness is only 0.148 s, 32.90% of its requests finish after their synthetic due times. The fixed-buffer policy is tardy on 94.29% of requests. Earliest and mean-only scheduling have zero tardy requests in this particular design. The contrast between mean lateness and tardy fraction shows that the chosen due windows place many occlusion-aware completions near their boundary. A study reporting only mean lateness would understate how often the scheduler crosses that boundary.

7 Sensitivity, Budgets, and Counterexamples

7.1 Uncertainty multiplier and scale mismatch

The multiplier sweep moves through a sequence of internally meaningful operating points. At zero, mean flow time is 1.896 s and the simulated violation share is 22.45% across the forty-run sweep. At one, those quantities are 2.054 s and 9.79%. At 1.64 they become 2.993 s and 2.86%, and at 2.8 they become 3.671 s and 0.10%. The difference between these values and the main table at the same multiplier follows from the smaller seed set. The sweep supports a monotone reduction of observed violation share on the tested grid, but supplies no proof that every intermediate multiplier or every alternative generator would behave monotonically.

Detour share is not monotone. It rises from 22.76% at zero to 62.97% at one, then falls to 9.95% at 2.33 before increasing again at 2.8. This behavior follows from discrete alternatives. At moderate conservatism, a detour can satisfy the margin immediately. At higher conservatism, even that route may fail during the occlusion interval, making a later direct approach less costly. At still higher conservatism, the post-occlusion baseline and phase can again make the detour useful. A plot showing only the multiplier and violation rate would miss this switch in mechanism.

Deliberately shrinking the assumed uncertainty scale to half its baseline produces 11.30% simulated violations and mean flow time 2.006 s. Increasing the multiplier on the scale to 1.5 produces 0.57% violations and mean flow time 3.590 s. The intermediate baseline case yields 2.86% and 2.993 s in this forty-run analysis. Because changing the scale is algebraically similar to changing k in the admission screen, these curves should be interpreted together. They do not provide two independent sources of evidence. Their different presentation emphasizes the practical distinction between selecting conservatism knowingly and acting with an incorrect uncertainty estimate.

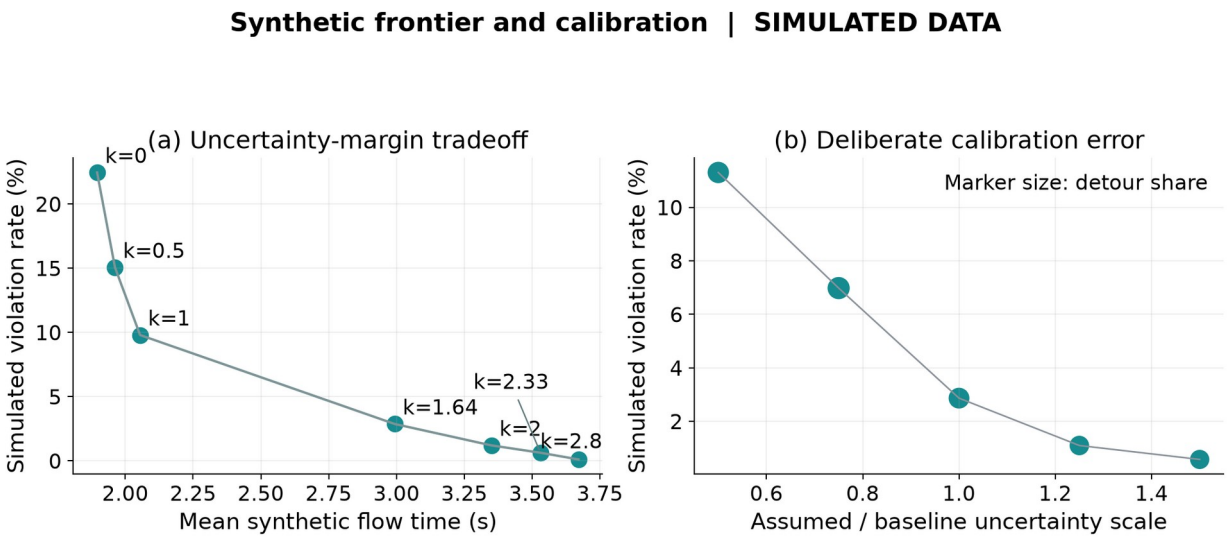


Figure 4. Simulated tradeoff and uncertainty-scale sensitivity using forty common-seed runs per setting. Panel (a) connects outcomes from seven uncertainty multipliers; the connecting segments are visual guides, not a proven continuous Pareto envelope. Panel (b) changes the assumed uncertainty scale while holding latent synthetic realizations fixed; marker size represents detour share.

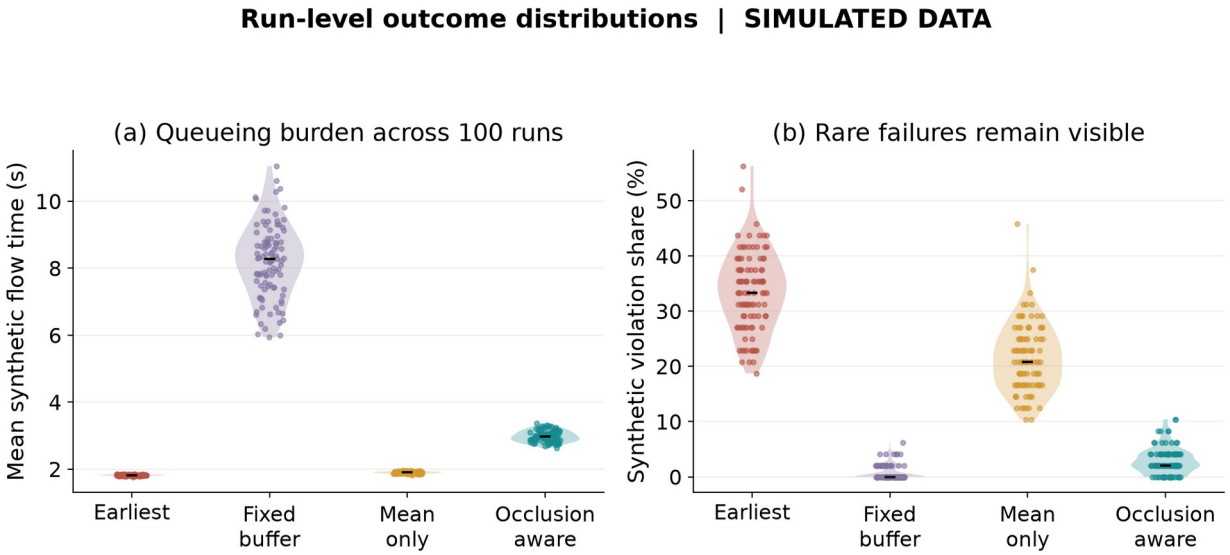


Figure 5. Simulated distributions of complete-run outcomes over 100 synthetic runs. Each dot represents a run mean or run violation share; translucent violins summarize their distribution and black marks indicate medians. Run-level plots preserve queue dependence and do not imply that individual requests are independent human-subject observations.

The apparent gains under larger uncertainty can also be misunderstood. The realized generator remains unchanged in the scale-mismatch sweep. Larger assumed uncertainty makes the scheduler alter its timing and route; it does not make the human trajectory less variable. If the real disturbance distribution broadened simultaneously with the assumed scale, the outcome could be very different. This separation between changing belief and changing the environment is central to interpreting any uncertainty-aware scheduling study. Our code exposes the distinction directly by applying the scale factor only to the forecast term, while generating realized perturbations from the baseline term.

7.2 Occlusion-duration sensitivity and search work

Increasing designed occlusion duration does not simply increase observed violation share in the adaptive scheduler. At half the baseline duration, mean flow time is 2.353 s and the violation share is 3.89%. At

twice the baseline duration, flow time grows to 4.546 s while violation share falls to 1.04%. The adaptive policy frequently waits beyond the longer uncertain interval, avoiding exposure at the cost of substantial delay. Detour share falls from 75.69% to 9.03% across these settings. The lower violation share under longer occlusion is therefore an action-mediated effect, not evidence that prolonged missing observations are intrinsically beneficial.

Table 3. Simulated occlusion-duration sensitivity over 30 common-seed runs per setting. All values refer to the occlusion-aware synthetic policy.

Duration multiplier	Flow time (s)	Wait (s)	Violation (%)	Detour (%)	Mean lateness (s)
0.50	2.353	0.275	3.89	75.69	0.012
1.00	3.007	1.043	3.06	43.06	0.151
1.50	3.769	1.880	1.53	21.74	0.623
2.00	4.546	2.701	1.04	9.03	1.276

Grid refinement lowers flow time in the twenty-run budget comparison, from 3.067 s at a 0.30 s spacing to 2.966 s at a 0.075 s spacing. Candidate-sample work rises from 384 to 1,464 evaluations per request. The finer grid's violation share is slightly higher, 3.23% rather than 2.71%. This is not paradoxical. The objective minimizes cost subject to the forecast margin, so a finer grid can locate cheaper candidates closer to the admission boundary. It is not optimizing the realized latent event count. Better resolution of a misspecified model need not improve every realized outcome.

Table 4. Simulated candidate-budget sensitivity over 20 common-seed runs per spacing; work counts exclude plotting and are not wall-clock benchmarks.

Grid spacing (s)	Candidates	Sample evaluations	Flow time (s)	Violation (%)	Mean lateness (s)
0.300	32	384	3.067	2.71	0.183
0.150	62	744	2.987	3.02	0.151
0.075	122	1464	2.966	3.23	0.142

A useful implementation budget must therefore include both search effort and a model-adequacy criterion. Increasing candidates can improve the discretized objective, but cannot repair an inaccurate uncertainty scale, an unmodeled bias, or a missing geometric constraint. It may even concentrate selections near precisely those boundaries where omitted terms matter most. The small arithmetic counts in this demonstration make that interaction easy to see. In a higher-dimensional planner, the same conceptual issue could coexist with expensive collision checks and estimator updates, making budget allocation a joint modeling and computation problem.

7.3 Constructed failure trajectories

The first illustrated failure is selected from the representative run, rather than chosen as an arbitrary dramatic curve. For its fourth request, the mean-only schedule encounters a realized downward clearance deviation during the unfavorable interval. The occlusion-aware schedule moves the approach later, after the designed visibility reset, and its realized surrogate remains above the threshold. Both traces use the same request-specific latent deviation and residual sequence. Their time axes differ because scheduling changes when the approach is executed. The shaded forecast bands are model displays, not confidence intervals estimated from repeated measurements of this request.

The second illustration is deliberately adversarial to the model. A single request is assigned a hidden-bias pulse amplitude of 0.58 m, far beyond the ordinary hidden-bias range. The occlusion-aware

scheduler admits an immediate detour whose forecast lower margin has minimum 0.208 m, above the 0.18 m threshold. The realized signed-clearance surrogate reaches approximately minus 0.112 m. This synthetic construction demonstrates a logical limitation: satisfying the stated forecast screen does not rule out an unmodeled displacement. The numerical amplitude is a counterexample design value, not a claim about plausible human movement.

Counterexamples are useful because aggregate performance can otherwise encourage an unjustified implication from small observed violation share to guaranteed operation. Even a setting with no failures in a finite synthetic sample would leave model discrepancy unresolved. Here the generator includes a modest bias component in ordinary runs and a much larger component in the constructed example. This makes the limitation visible at two levels: observed residual failures within the designed comparison, and failure under an explicitly altered disturbance outside it. Neither can be addressed merely by attaching a more optimistic name to the constraint.

8 Interpretation and Limits of the Demonstration

The results suggest a specific systems mechanism under the declared assumptions. Nominal screening removes some poor approaches, uncertainty screening often substitutes a detour or waits for a visibility reset, and unconditional buffering can overload a queue whose arrival spacing was designed for faster service. These statements concern a synthetic process with known occlusion windows, a scalar clearance coordinate, and prescribed phases. They do not establish a new sensing method or a new physical handover controller. The strongest defensible conclusion is that timing, route cost, and information assumptions should be examined jointly when presenting a scheduling rule.

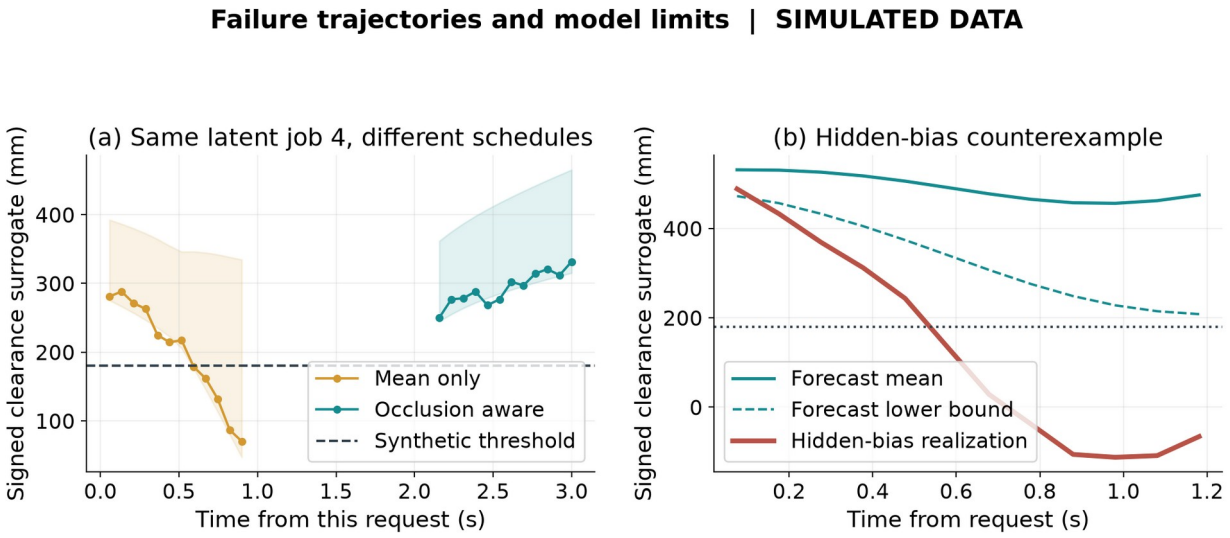


Figure 6. Simulated failure trajectories. Panel (a) compares mean-only and occlusion-aware schedules for the same latent fourth request in seed 73219; scheduling changes the times at which synthetic clearance is evaluated. Panel (b) is a constructed out-of-model hidden-bias counterexample: an admitted synthetic detour satisfies the forecast margin but violates the realized signed-clearance threshold. Negative surrogate values denote abstract penetration, not negative physical distance.

Several omissions would matter before any real-world study. Human motion is represented by a fixed nominal function and a small number of random variables. The model does not include multimodal reach intent, response to robot motion, startle, hesitation, fatigue, object affordances, or changes of task. A person can react to the robot, so a counterfactual schedule may change the human trajectory itself. Our common-latent comparison assumes that it does not. The resulting paired differences are useful for controlled simulator analysis but cannot be interpreted as causal estimates of how people would respond.

The nominal detour is also a scalar abstraction. It adds clearance and duration without checking joint limits, self-collision, reachability, orientation, or swept volume. The workspace drawing supports explanation only. A real detour might increase clearance from a hand while decreasing clearance from a forearm or another object. It might also change the receiver's reachable grasp configuration. Whole-body geometric validation would therefore require replacing the scalar margin with an appropriate scene model and retaining the uncertainty and timing considerations at that higher level of detail.

The visibility reset is particularly favorable to waiting. The scheduler knows when the interval ends, and the uncertainty growth component vanishes immediately at that point. In reality, reacquisition can be delayed, biased, or ambiguous. A hand may remain hidden behind the object, and a newly visible detection may not refer to the expected limb. Making visibility recovery uncertain would broaden the decision problem from deterministic interval avoidance to contingency planning. A stop, query, or withdrawal action could become preferable to waiting for a forecast that never arrives. None of those actions is represented by the maximum-margin fallback in the current code.

The fixed-buffer comparison deserves equally careful interpretation. Its congestion is intentionally induced by applying an additional delay after availability on every request. This is a valid synthetic policy but not a strong benchmark for all conservative scheduling. A conditional buffer, background sensing during unrelated work, dynamic task reordering, or a queue-aware rule could improve its performance. The present comparison isolates a mechanism rather than exhaustively ranking the scheduling literature. More complex baselines would be meaningful only after the task and information assumptions were made correspondingly richer.

A credible next research stage would begin with measurement and model criticism. It would define the actual precontact event, identify which robot and human body geometries matter, record visibility interruptions, quantify reacquisition behavior, and distinguish forecast error from execution tracking error. It would also specify what the robot does when no candidate is acceptable. Only then would fitted distributions, model coverage, and a real-time implementation support claims about an operational setting. Human acceptability would require its own study design and measures, rather than being inferred from lower surrogate violation rates or shorter synthetic completion times.

9 Conclusion

This synthetic study makes a handover scheduling dilemma concrete through a transparent event model and finite candidate search. Across one hundred simulated runs, the occlusion-aware policy reduces sampled surrogate threshold violations relative to mean-only scheduling while increasing request-to-completion time. An unconditional buffer achieves a lower simulated violation share at a much larger queueing cost in the deliberately congested scenario. Sensitivity analyses show that route use can change nonmonotonically, longer occlusion can lower observed violations through increased waiting, and finer search can improve modeled cost without reducing realized failures.

The central lesson is about the relationship between a decision rule and its assumptions. An uncertainty margin is informative only to the extent that its scale, temporal event, and omitted disturbances are understood. The supplied counterexample exhibits failure despite candidate admission, and the simulator supplies no physical safety guarantee. The study and accompanying reproducible assets use entirely synthetic data. Their value lies in the auditability of the reasoning and the visibility of its limits, rather than evidence of performance on robots, people, or a trained-model benchmark.

References

[1] V. Ortenzi, A. Cosgun, T. Pardi, et al., “Object handovers: a review for robotics,” IEEE Transactions on Robotics, 37(6), pp. 1855–1873, 2021. doi: 10.1109/TRO.2021.3075365.

[2] K. Strabala, M. K. Lee, A. Dragan, et al., “Towards Seamless Human-Robot Handovers,” Journal of Human-Robot Interaction, 2(1), pp. 112–132, 2013. [Online]. Available: https://publications.ri.cmu.edu/towards-seamless-human-robot-handovers.

[3] M. Cakmak, S. S. Srinivasa, M. K. Lee, et al., “Human Preferences for Robot-Human Hand-over Configurations,” IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1986–1993, 2011. [Online]. Available: https://publications.ri.cmu.edu/human-preferences-for-robot-human-hand-over-configurations.

[4] A. D. Dragan, K. C. T. Lee, S. S. Srinivasa, “Legibility and Predictability of Robot Motion,” ACM/IEEE International Conference on Human-Robot Interaction, pp. 301–308, 2013. [Online]. Available: https://publications.ri.cmu.edu/legibility-and-predictability-of-robot-motion.

[5] A. Kupcsik, D. Hsu, W. S. Lee, “Learning Dynamic Robot-to-Human Object Handover from Human Feedback,” arXiv preprint arXiv:1603.06390 (ISRR 2015 author manuscript), 2016. [Online]. Available: https://arxiv.org/abs/1603.06390.

[6] P. Marra, S. Berti, G. M. Caddeo, et al., “Temporal Tactile Encoding and Compliance for Intent-Aware Robot-to-Human Bimanual Handover,” arXiv preprint arXiv:2609.05282v2, 2026. doi: 10.48550/arXiv.2609.05282.

[7] J. Shah, J. Wiken, B. C. Williams, et al., “Improved human-robot team performance using Chaski, a human-inspired plan execution system,” ACM/IEEE International Conference on Human-Robot Interaction, 2011. [Online]. Available: https://www.media.mit.edu/publications/improved-human-robot-team-performance-using-chaski-a-human-inspired-plan-execution-system/.

[8] S. Nikolaidis and J. A. Shah, “Human-robot cross-training: Computational formulation, modeling and evaluation of a human team training strategy,” ACM/IEEE International Conference on Human-Robot Interaction, pp. 33–40, 2013. doi: 10.1109/HRI.2013.6483499.

[9] R. Dechter, I. Meiri, J. Pearl, “Temporal constraint networks,” Artificial Intelligence, 49(1–3), pp. 61–95, 1991. doi: 10.1016/0004-3702(91)90006-6.

[10] J. F. Fisac, A. Bajcsy, S. L. Herbert, et al., “Probabilistically Safe Robot Planning with Confidence-Based Human Predictions,” Robotics: Science and Systems; arXiv:1806.00109, 2018. doi: 10.48550/arXiv.1806.00109.

[11] L. Blackmore, M. Ono, B. C. Williams, “Chance-Constrained Optimal Path Planning With Obstacles,” IEEE Transactions on Robotics, 27(6), pp. 1080–1094, 2011. doi: 10.1109/TRO.2011.2161160.

[12] M. Prandini, J. Hu, J. Lygeros, et al., “A Probabilistic Approach to Aircraft Conflict Detection,” IEEE Transactions on Intelligent Transportation Systems, 1(4), pp. 199–220, 2000. [Online]. Available: https://prandini.faculty.polimi.it/html/papers.html.

[13] A. D. Ames, X. Xu, J. W. Grizzle, et al., “Control Barrier Function Based Quadratic Programs for Safety Critical Systems,” IEEE Transactions on Automatic Control, 62(8), pp. 3861–3876, 2017. doi: 10.1109/TAC.2016.2638961.

[14] J. Zeng, B. Zhang, K. Sreenath, “Safety-Critical Model Predictive Control with Discrete-Time Control Barrier Function,” arXiv preprint arXiv:2007.11718, 2020. [Online]. Available: https://arxiv.org/abs/2007.11718.

[15] T. Koller, F. Berkenkamp, M. Turchetta, et al., “Learning-based Model Predictive Control for Safe Exploration,” IEEE Conference on Decision and Control; arXiv:1803.08287, 2018. doi: 10.48550/arXiv.1803.08287.

[16] S. L. Herbert, M. Chen, S. Han, et al., “FaSTrack: a Modular Framework for Fast and Guaranteed Safe Motion Planning,” IEEE Conference on Decision and Control, 2017. doi: 10.1109/CDC.2017.8263867.

[17] S. Karaman and E. Frazzoli, “Sampling-based Algorithms for Optimal Motion Planning,” The International Journal of Robotics Research, 30(7), pp. 846–894, 2011. [Online]. Available: https://dspace.mit.edu/entities/publication/77dff1d1-ffd6-42f3-a3a5-fef87a964959.

[18] R. G. Sargent, “Verification and validation of simulation models,” Journal of Simulation, 7(1), pp. 12–24, 2013. doi: 10.1057/jos.2012.20.

[19] G. C. Calafiore and M. C. Campi, “The Scenario Approach to Robust Control Design,” IEEE Transactions on Automatic Control, 51(5), pp. 742–753, 2006. doi: 10.1109/TAC.2006.875041.

[20] M. C. Campi and S. Garatti, “The Exact Feasibility of Randomized Solutions of Uncertain Convex Programs,” SIAM Journal on Optimization, 19(3), pp. 1211–1230, 2008. doi: 10.1137/07069821X.