Autonomous Navigation and Obstacle Avoidance for Industrial Robots in Unstructured Construction Sites

Rohan Conaty
Department of Electrical Engineering and Computer Science, University of Kansas, Lawrence, KS, USA.
conatyrohan@ku.edu

Maanas Rama
Department of Computer Science and Engineering, University at Buffalo, Buffalo, NY, USA.
mrama@buffalo.edu

Abstract

Industrial robots are increasingly expected to operate beyond fenced factory cells, where tasks involve moving through construction sites that are only partially known, geometrically irregular, and populated by workers, materials, and equipment. The central problem is not merely obstacle detection; it is the coupling of uncertain site representation, construction-specific traversability, dynamic obstruction, and hard safety constraints under changing operational conditions. This manuscript develops an integrated methodological framework for autonomous navigation and obstacle avoidance for industrial robots in unstructured construction sites. The framework represents the site as layered probabilistic maps of occupancy, traversability, semantics, and uncertainty; computes a risk-sensitive traversability functional; organizes motion through a hierarchical global-local planning architecture; and enforces robust clearance and control barrier safety constraints. The contribution is a coherent formulation that connects uncertainty-aware mapping to constrained trajectory generation and commissioning logic rather than a single isolated algorithm. Analytical assessment addresses perception degradation, map staleness, planning latency, dynamic obstacle prediction, and the trade-off between conservative safety margins and productive motion. The formulation also clarifies where factory assumptions break down, including wheel-terrain interaction, overhead obstruction, temporary structures, and mixed human-robot traffic. By making these dependencies explicit, the manuscript provides a technically defensible basis for designing and evaluating site-capable navigation systems without treating construction sites as scaled versions of factory floors.

Keywords

construction robotics; autonomous navigation; obstacle avoidance; uncertain occupancy mapping; safety-constrained planning; unstructured environments

1. Introduction

Construction robotics has developed through prefabrication, bricklaying, concrete distribution, and on-site assembly systems that reduce manual effort and improve repeatability [1]. More recent work has broadened the scope from fixed automation to mobile and task-configurable systems that must tolerate the physical variability of construction sites [2]. This shift changes the navigation problem. A factory robot moves through a mapped corridor with known floor properties, stable obstacles, and controlled human access. A construction robot must move through temporary layouts, incomplete structures, material staging areas, floor

transitions, and work zones whose geometry and occupancy change within a shift. The resulting problem combines probabilistic perception, nonuniform terrain, semantic hazard interpretation, dynamic obstruction, and industrial safety constraints.

Autonomous navigation under uncertainty rests on probabilistic state estimation and mapping [3], occupancy grid representations [4], and planning algorithms that reason over configuration or state space [5]. These foundations remain necessary, but they are not sufficient for construction sites. Occupancy grids usually encode geometric occupancy rather than load-bearing capacity, trip hazards, wet concrete, exposed reinforcement, or temporary supports. Planning algorithms assume that the cost function and constraint set are meaningful representations of the vehicle's feasible and safe motion. On a site, both the cost function and the feasible set are partially observed and can change faster than a static map can be updated. A robot that treats all unobserved space as free will collide with material stacks or enter unsafe zones; a robot that treats all unobserved space as occupied will remain immobilized.

The central contribution of this manuscript is an integrated methodological framework for site-capable navigation and obstacle avoidance. The framework has four connected functions: a layered probabilistic representation that separates occupancy, traversability, semantics, and uncertainty; a risk-sensitive traversability functional that converts map layers into motion costs while preserving hard safety constraints; a hierarchical planner that combines global route search with short-horizon local optimization; and a safety filter that projects nominal commands onto a certified or analytically bounded safe set. The manuscript develops the governing relationships for these functions and assesses their design trade-offs. It does not report a field trial or benchmark result. Instead, it provides a rigorous analytical account of how perception uncertainty, construction semantics, planning latency, and safety margins interact in unstructured sites.

2. Domain Problem and Formalization

2.1 Unstructured construction sites as navigation environments

Construction sites are not unstructured in the sense of lacking pattern. They are semi-structured and temporally ordered. Structural grids, haul routes, and staging areas impose regularities, but these regularities are interrupted by temporary equipment, material deliveries, scaffolding, and work-front changes. Obstacles therefore fall into four operationally distinct classes. Static structural obstacles include columns, walls, openings, and permanent services. Semi-static obstacles include pallets, formwork, rebar bundles, and generators that move occasionally but are stationary during a navigation episode. Dynamic obstacles include workers, lift equipment, and vehicles whose future positions are uncertain. Soft or semantic hazards include unstable ground, wet surfaces, overhead clearance limits, and restricted access zones that may not appear as geometric obstacles.

This classification matters because a single geometric occupancy map collapses distinct risks into one representation. A pallet and a column may both be occupied cells, but their consequences and update rates differ. A column is stable and can be mapped with high confidence. A pallet can be moved without notice, so its map evidence decays. A worker is dynamic and requires prediction. Wet concrete may be traversable in a physical sense but unacceptable for an industrial robot with a heavy tool payload. Navigation therefore requires a representation that preserves the type and freshness of evidence, not only its occupancy probability.

2.2 Formal problem statement

Let the robot state at discrete time t be x_t in a state space X, and let the applied control be u_t in an admissible set U. The site map is a time-varying random field m_t that contains occupancy, traversability, and semantic attributes. The robot dynamics and sensing model are written as

$$x_{t+1} = f(x_t, u_t) + w_t, z_t = g(x_t, m_t) + v_t$$

where f is the motion model, g is the observation model, z_t is the sensor measurement, and w_t and v_t are process and measurement disturbances. The disturbances are not assumed to be Gaussian in all conditions; they summarize unmodeled wheel slip, sensor noise, calibration error, and map registration error. The map m_t includes a set of obstacles O_j(t), each with an estimated pose, extent, uncertainty, and, where available, a semantic class. The navigation task is to generate a control sequence that reaches a task goal while maintaining clearance, stability, and permission constraints under the uncertain map and predicted obstacle motion.

The problem is partially observable because neither the current map nor the future obstacle motion is known exactly. A practical policy must therefore use the belief over x_t and m_t rather than a single state estimate. This requirement is especially strong in construction because map evidence can be stale. A corridor may have been clear when observed and blocked later by a delivery. The belief must encode not only where obstacles are believed to be but also how recently and reliably that belief was established.

2.3 Limits of existing navigation components and site constraints

Reactive collision avoidance methods provide fast responses but limited foresight. The dynamic window approach evaluates admissible velocity commands over a short horizon and selects commands that avoid obstacles while respecting robot kinematics [6]. It is effective in local clutter but can become trapped when a temporary obstruction blocks the only local passage. Artificial potential fields provide continuous repulsive action around obstacles [7], and their dynamic extensions improve motion planning in changing environments [10]. Their known weakness is local minima and oscillatory behavior in narrow or nonconvex passages, which are common around scaffolding and material stacks.

Control barrier functions offer a principled way to enforce safety for a known control-affine model by rendering a safe set forward invariant [8]. They are attractive for industrial robots because they can filter nominal commands and respect actuator limits. Their guarantee depends on model fidelity and on the existence of a valid barrier function for the relevant obstacle geometry. Integrated navigation systems combine mapping, planning, and control in a modular stack [9], but the default assumptions of such stacks often favor indoor or road-like environments. Construction sites introduce nonplanar ground, low-clearance overhead structures, reflective or dusty surfaces, and frequent changes in the semantic meaning of space. Industrial robots designed for factory operation also carry tooling, cable management, and counterweights that alter their footprint and braking behavior when transferred to site conditions [11]. These constraints require a navigation architecture that treats safety as a set of explicit constraints rather than a penalty term buried in a cost function.

3. Methodological Framework

3.1 Multimodal site representation and uncertainty propagation

The representation layer should be multimodal because no single sensor provides reliable evidence across construction conditions. Three-dimensional lidar supports geometric mapping

and obstacle segmentation. Stereo or depth cameras provide dense local geometry and semantic cues. Inertial and wheel-odometry measurements constrain short-term motion. Ultra-wideband or site-based localization can bound drift when geometric features are sparse. Construction robot architectures must also accommodate task-specific tooling and changing bases, which affects sensor placement and calibration [12]. Point-cloud registration aligns successive scans and supports map maintenance [13], while lidar odometry and mapping provide continuous pose estimates in geometrically rich environments [14]. Neither method removes the need for uncertainty propagation, because registration can fail in repetitive or feature-poor areas such as unfinished corridors.

For occupancy, a log-odds update provides a transparent way to accumulate evidence while retaining the ability to decay stale information. Let L_t(o) be the log-odds that cell o is occupied after measurements up to time t, and let L_0(o) be the prior log-odds for that cell. The update is

$$L_t(o) = L_{t-1}(o) + \log\left(p\frac{(o|z_t)}{(1-p(o|z_t))}\right) - L_0(o)$$

where p(o|z_t) is the inverse sensor model probability that cell o is occupied given measurement z_t. The subtraction of the prior prevents repeated application of the same prior during recursive updates. The conditional independence assumption behind this recursion is only an approximation, especially when dust, smoke, or reflective surfaces produce correlated errors across neighboring cells. The framework therefore treats L_t(o) as one evidence channel and maintains a separate uncertainty field. The uncertainty field can be derived from the variance of registration residuals, the age of the last observation, and the local density of valid returns. This separation prevents a high-confidence occupancy estimate from being mistaken for a high-confidence traversability judgment.

The semantic layer assigns each observed region a class such as structural, material, mobile equipment, human work zone, or restricted access. Class probabilities should be stored with the same spatial indexing as occupancy so that a planner can query both geometric clearance and semantic permission. A cell may be geometrically free but semantically forbidden because it lies under a suspended load or within an exclusion zone. Conversely, a cell may be geometrically occupied by a temporary barrier that can be removed by a site operator, which justifies a different replanning response than a structural column.

3.2 Traversability and semantic risk assessment

Traversability is not equivalent to free space. An industrial robot with a narrow wheelbase and a high center of mass may be unable to cross a slope or a floor transition that a small mobile robot would ignore. The traversability cost for a candidate pose x can be written as

$$T(x) = \alpha T_r(x) + \beta T_s(x) + \gamma T_h(x) + \lambda U(x)$$

where T_r is a terrain-roughness cost derived from local surface residuals, T_s is a slope and step cost, T_h is a semantic hazard cost, and U is a map-uncertainty cost. The coefficients α, β, γ, and λ set the relative influence of these terms and should be tied to robot stability margins and task sensitivity rather than tuned as arbitrary weights. Lower T indicates lower cost. Roughness captures small-scale irregularities that increase vibration or wheel slip. Slope and step terms capture geometric conditions that can cause loss of traction or tipping. Semantic hazard captures restricted zones, fragile surfaces, and overhead clearance limits. Uncertainty penalizes motion through regions where the map is unreliable.

The cost functional is deliberately separated from hard constraints. A high semantic hazard may make a region costly, but a physical clearance limit must remain a constraint. If all safety requirements are folded into T, the optimizer can trade safety for progress when the cost landscape favors a shorter route. The framework therefore uses T for ranking and smoothing feasible trajectories, while clearance, stability, and permission limits are enforced as constraints in the local planner and safety filter.

3.3 Hierarchical planning and obstacle avoidance

The planner should be hierarchical because global route selection and local obstacle avoidance operate on different time scales. The global layer searches over a layered costmap that combines T, semantic restrictions, and expected travel time. Graph search methods such as A and Dijkstra provide deterministic optimality for nonnegative edge costs and remain useful when the costmap is stable and the graph is modest in size [16], [17]. Sampling-based planners provide probabilistic completeness and asymptotic optimality under suitable conditions and are better suited to high-dimensional or nonconvex spaces [15]. A practical architecture can use graph search for route-level decisions and sampling or optimization for local motion when the robot encounters clutter that invalidates the route.

The local planner solves a short-horizon optimization over a control sequence u = (u_t, ..., u_{t+H−1}). A representative objective is

where H is the planning horizon, γ is a discount factor in (0, 1], λ penalizes control changes to reduce wear and oscillation, and Φ(x_f) is a terminal cost that rewards progress toward the goal. The sequence is subject to the motion model in the first equation, actuator limits, and clearance constraints. The horizon must be long enough to anticipate braking and turning but short enough to keep computation within the control period. A long horizon improves foresight but amplifies model error and map staleness; a short horizon reacts quickly but can enter dead ends. The hierarchical structure mitigates this trade-off by letting the global layer maintain route-level progress while the local layer handles immediate obstruction.

Clearance should be robust rather than nominal. For obstacle O_j with estimated position uncertainty σ_j, a conservative constraint is

$$d(x_k, O_j) \geq d_{min} + \kappa\sigma_j + ||v||\Delta t$$

where d is the signed distance from the robot body to the obstacle, d_min is the required physical clearance, κ is a confidence multiplier, v is the robot velocity, and Δt is the planning interval used to account for stopping distance. The term κσ_j expands the obstacle according to position uncertainty. The velocity term accounts for the distance traveled before a new command can be applied. This constraint is intentionally conservative. If σ_j is large because the obstacle has not been observed recently, the planner will prefer a wider berth or a slower approach. The constraint does not guarantee safety by itself, because it relies on the accuracy of d and σ_j, but it makes the source of conservatism explicit and adjustable.

3.4 Safety-constrained motion control

The local planner produces a nominal command, but the final command should pass through a safety filter that enforces the tightest instantaneous constraint. Control barrier functions provide a suitable formal structure. Let h(x) be a continuously differentiable function whose nonnegative superlevel set defines safe states, so that h(x) ≥ 0 implies the

robot maintains required clearance from the nearest obstacle or forbidden region. A common condition is

$$\dot{h}(x,u) + \alpha h(x) \geq 0$$

where $\dot{h}$ is the time derivative of h along the controlled dynamics and α is a positive constant. If the nominal command would violate this condition, the safety filter solves a minimally invasive quadratic program that finds the closest admissible command satisfying the barrier condition and actuator bounds. The filter is attractive because it can be added to an existing planner and because it responds at the control rate. Its guarantee depends on the accuracy of the control-affine model and on the existence of an admissible command near the boundary. Saturation, unmodeled wheel slip, and sudden appearance of an obstacle can invalidate the guarantee. The barrier condition should therefore be paired with conservative clearance margins, speed limits near occlusion boundaries, and a fail-safe stop when sensor health or localization confidence falls below a threshold. Collision-free navigation algorithms provide a broader repertoire of strategies, including reactive, geometric, and optimization-based methods [18], while industrial robot control design supplies the actuator and stability constraints that the filter must respect [19].

4. Analytical Assessment and Design Trade-offs

4.1 Perception degradation and map staleness

Construction conditions degrade perception in predictable ways. Dust and airborne particles scatter lidar returns and reduce visible range. Direct sunlight and shadows affect stereo depth. Reflective sheeting and wet surfaces produce invalid or displaced returns. Rebar, scaffolding, and mesh create thin obstacles that may be missed at range and then appear abruptly. The occupancy update in the second equation will accumulate evidence inconsistently under these conditions because the inverse sensor model is not stationary. A dust cloud can generate spurious occupied cells that persist after the cloud disperses. A reflective surface can generate free-space evidence behind a real obstacle.

The framework responds with two mechanisms. First, the uncertainty field inflates cells whose evidence is old, sparse, or inconsistent across sensors. Second, the planner treats high-uncertainty space as costly rather than forbidden unless a semantic rule forbids it. This allows the robot to proceed cautiously through uncertain free space while avoiding overreaction to transient sensor artifacts. The design trade-off is that excessive inflation reduces the set of traversable routes and may cause the robot to wait for operator intervention. The correct balance depends on the cost of delay relative to the consequence of a collision. On a site with human workers, the consequence is asymmetric, so the uncertainty penalty should be higher near work zones and lower in closed staging areas.

4.2 Planning latency and dynamic obstacle response

Dynamic obstacle response is limited by the total latency from sensing to actuation. If a worker enters the path, the robot must detect the worker, update the map or tracker, replan or filter the command, and execute the new command. Each stage consumes time. A short local horizon reduces computation but may not anticipate a worker emerging from behind a material stack. A long horizon improves anticipation but may rely on an obstacle prediction that becomes invalid when the worker changes direction. The robust clearance constraint provides a partial answer by expanding obstacles according to uncertainty and stopping distance, but it cannot compensate for a detection that occurs too late.

A practical design should separate detection latency from response latency. The safety filter should operate on the fastest reliable obstacle channel, such as a forward depth or lidar region, and should be able to stop or slow the robot without waiting for a full map update. The local planner can then update the route at a lower rate. This two-rate architecture reduces the chance that map maintenance delays a safety response. It also creates a hierarchy of evidence: immediate geometric obstacles trigger the safety filter, while semantic and route-level changes trigger replanning.

4.3 Safety-productivity trade-offs

Every safety margin reduces the set of admissible motions. A larger clearance margin protects against localization error and obstacle uncertainty, but it can make a narrow passage infeasible or force a longer route. A lower speed reduces stopping distance, but it increases task time and may block other site traffic. The analytical framework makes these trade-offs explicit through d_{min}, κ, σ_j, and the velocity term in the clearance constraint. Increasing κ or d_{min} produces a more conservative policy; decreasing them increases productivity but shifts risk to model accuracy and reaction time. The correct setting is not a universal constant. It should depend on the robot's braking capability, the consequence of contact, the reliability of localization, and the density of human traffic. In mixed traffic, the margin should be asymmetric and larger in the direction of motion than behind the robot.

4.4 Calibration, commissioning, and assurance

A site-capable navigation system requires calibration that goes beyond factory acceptance. Sensor extrinsics must be verified after tooling changes, because a new implement can alter the robot's footprint and occlude sensors. Point-cloud registration and absolute orientation methods support geometric alignment and extrinsic estimation [20]. Localization drift must be monitored during operation, because a slowly growing pose error can invalidate the clearance constraint even when the obstacle map is correct. Commissioning should therefore include tests that deliberately introduce map staleness, occlusion, and dynamic obstacles, and should record whether the safety filter or the planner is responsible for each stop or slowdown. This separation is important for assurance: a stop caused by a conservative uncertainty margin is different from a stop caused by a detected obstacle, and the two should not be conflated in operational logs.

5. Implementation Implications

The framework implies that site navigation should be designed around evidence quality rather than map completeness. A robot does not need a perfect global map to move safely, but it does need a reliable local belief, a conservative response to uncertainty, and a safety filter that acts on the fastest available obstacle evidence. Site planning can support this behavior by defining designated haul routes, staging areas, and exclusion zones that reduce the semantic complexity of the navigation problem. Temporary signage and site-localization infrastructure can reduce uncertainty in areas where geometric features are sparse. These are not substitutes for onboard perception, but they lower the burden on perception and make the robot's behavior more predictable to human workers.

The architecture also has implications for robot design. A robot with a large turning radius, limited ground clearance, or slow braking will require wider margins and longer planning horizons than a compact platform. Tooling that projects beyond the base must be included in the collision model. Sensor placement must balance field of view against dust,

vibration, and impact risk. These constraints should be treated as co-design variables rather than fixed parameters chosen after the navigation stack is assembled.

6. Conclusion

This manuscript has developed an integrated methodological framework for autonomous navigation and obstacle avoidance by industrial robots in unstructured construction sites. The framework separates occupancy, traversability, semantics, and uncertainty so that geometric free space is not mistaken for safe or permissible space. It converts these layers into a risk-sensitive traversability cost while retaining clearance, stability, and permission as hard constraints. A hierarchical planner combines global route search with short-horizon local optimization, and a control barrier safety filter enforces instantaneous safety on the nominal command. The analytical assessment shows that the main design tensions arise from perception degradation, map staleness, planning latency, and the productivity cost of conservative margins. The formulation does not eliminate these tensions, but it makes their sources explicit and connects them to adjustable design parameters. That connection is the practical value of the framework: it allows site navigation to be reasoned about as a joint problem of uncertain representation, constrained motion, and commissioning rather than as a collection of isolated obstacle-avoidance modules.

References

1. Bock, T. (2007). Construction robotics. Autonomous Robots, 22(3), 201–209.
2. Saidi, K. S., Bock, T., & Georgoulas, C. (2016). Robotics in construction. In B. Siciliano & O. Khatib (Eds.), Springer handbook of robotics (2nd ed., pp. 1493–1520). Springer.
3. Thrun, S., Burgard, W., & Fox, D. (2005). Probabilistic robotics. MIT Press.
4. Elfes, A. (1989). Using occupancy grids for mobile robot perception and navigation. Computer, 22(6), 46–57.
5. LaValle, S. M. (2006). Planning algorithms. Cambridge University Press.
6. Fox, D., Burgard, W., & Thrun, S. (1997). The dynamic window approach to collision avoidance. IEEE Robotics & Automation Magazine, 4(1), 23–33.
7. Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. International Journal of Robotics Research, 5(1), 90–98.
8. Ames, A. D., Coogan, S., Egerstedt, M., Notomista, G., Sreenath, K., & Tabuada, P. (2019). Control barrier functions: Theory and applications. In 2019 European Control Conference (ECC) (pp. 3420–3431). IEEE.
9. Macenski, S., Martín, F., White, R., & Clavero, J. G. (2020). The Marathon 2: A navigation system. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 2718–2725). IEEE.
10. Ge, S. S., & Cui, Y. J. (2002). Dynamic motion planning for mobile robots using potential field method. Autonomous Robots, 13(3), 207–222.
11. Luo, D., & Yu, L. (2021). From factory to site—designing for industrial robots used in on-site construction. In Automating cities: design, construction, operation and future impact (pp. 87-109). Singapore: Springer Singapore.

12. Bock, T., & Linner, T. (2016). Construction robots: Elementary technologies for automated building construction. Cambridge University Press.

13. Besl, P. J., & McKay, N. D. (1992). A method for registration of 3-D shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence, 14(2), 239–256.

14. Zhang, J., & Singh, S. (2014). LOAM: Lidar odometry and mapping in real-time. In Robotics: Science and Systems (Vol. 2, No. 9, pp. 1–9).

15. Karaman, S., & Frazzoli, E. (2011). Sampling-based algorithms for optimal motion planning. International Journal of Robotics Research, 30(7), 846–894.

16. Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. IEEE Transactions on Systems Science and Cybernetics, 4(2), 100–107.

17. Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. Numerische Mathematik, 1, 269–271.

18. Hoy, M., Matveev, A. S., & Savkin, A. V. (2015). Algorithms for collision-free navigation of mobile robots in complex cluttered environments: A survey. Robotica, 33(3), 463–497.

19. Siciliano, B., Sciavicco, L., Villani, L., & Oriolo, G. (2010). Robotics: Modelling, planning and control. Springer.

20. Horn, B. K. P. (1987). Closed-form solution of absolute orientation using unit quaternions. Journal of the Optical Society of America A, 4(4), 629–642.