From Sensing to Scheduling: Deviation-Based Framework for Sensor-Aware Progress Monitoring and Schedule Control

Shabtai Isaac[1] and Gunnar Lucko[2]

[1] Department of Structural Engineering, Ben-Gurion University of the Negev, Beersheba, Israel
[2] Department of Civil and Environmental Engineering, Catholic University of America, Washington, DC, USA
isaacsh@bgu.ac.il, lucko@cua.edu

Abstract –

Automated progress monitoring has advanced through reality capture (images, laser scanning, SLAM/mobile mapping) and data-driven interpretation aligned to BIM/4D models. Yet many workflows still stop at percent-complete estimates and leave the schedule-update step under-specified, especially when completion depends on inspections, tests, and formal acceptance. This paper proposes a management-oriented sensing-to-scheduling method that makes that step explicit and computable. First, it separates production-facing progress monitoring from schedule control, highlighting their different units (work vs. time) and decision contexts. Second, it introduces sensor-aware activity scheduling based on evidence: Existence evidence from geometry/vision and performance/acceptance evidence from QA tests, inspections, and approvals. Third, it defines a translation layer that converts measured work quantities into work backlog/forelog, time delay, forecasted finishes, and recovery targets (required productivity) suitable for schedule updating. The approach is designed to complement existing 4D BIM and digital twin pipelines rather than replace them. A small illustrative example demonstrates the calculations and shows how acceptance gates remain explicit milestones. The discussion clarifies applicability, limitations, and research needs for fusing QA data and handling uncertainty.

Keywords –

Construction automation, progress monitoring, schedule control, backlog, 4D BIM, digital twin, sensor-aware WBS

1 Introduction

Automation has made construction sites increasingly measurable. Reality capture and sensing – laser scanning, mobile mapping/simultaneous localization and mapping (SLAM) devices, drones, and wearable or fixed cameras – can provide frequent, spatially rich records of as-built conditions that can be aligned to building information models (BIM) and four-dimensional (4D) schedules. Early work demonstrated that fusing three-dimensional (3D) laser scans with 4D information enables automated structural progress tracking at accuracy comparable to typical manual updates [1]. More recent research has advanced activity-level interpretation of visual data, including methods that estimate activity-wise percent complete from images aligned to a 4D BIM [2]. Systematic reviews show that the technical pipeline for computer-vision-based progress monitoring (data acquisition, information retrieval, progress estimation, and visualization) is maturing rapidly [3].

However, practical schedule control still hinges on a step that is frequently implicit: Converting measured physical progress into schedule updates and intervention targets that are defensible under real project constraints. Many workflows stop at percent-complete estimates and leave the ‘update the schedule’ step to ad-hoc interpretation, even though schedule status often depends on inspections, tests, and formal acceptance that are not purely visual. This paper addresses that gap with a management-oriented sensing-to-scheduling method that remains realistic about what sensing can and cannot verify.

1.1 Objectives and Contributions

The paper has three objectives:

- Clarify the conceptual mismatch between production-facing progress monitoring and schedule control.
- Introduce sensor-aware activity and work breakdown structure (WBS) authoring based on evidence, distinguishing existence evidence from acceptance evidence.

- Define a computable translation layer that converts measured work into backlog/forelog, delay, forecast finishes, and recovery targets suitable for schedule updating.

The primary contributions are methodological:

- An explicit separation of progress monitoring and schedule control as distinct (but coupled) control loops.
- A sensor-aware WBS authoring concept that represents non-visual acceptance gates as first-class schedule milestones.
- A deviation-based translation that links measured work quantities to time-based schedule implications and decision targets, designed to complement 4D BIM and digital twin pipelines.

2 Background and Related Works

Related work most directly relevant to this paper can be organized into three streams. The first stream is automated progress monitoring, where images, video, laser scans, and other sensing modalities are used to detect installed work and estimate production status. A systematic review by Rehman et al. organizes computer-vision-based progress monitoring into data acquisition, information retrieval, progress estimation, and output visualization, and shows that automation is advancing but remains uneven across the full workflow [3]. Representative studies include Turkan et al., who fused 4D schedules and 3D sensing for automated structural progress tracking [1], Pal et al., who estimated activity-level percent complete from images aligned to 4D BIM [2], and Zhao et al., who inferred task-level progress from indoor positioning data [4]. Collectively, this stream strengthens the measurement of executed work but often stops at status estimation or percent-complete reporting.

The second stream is BIM/4D and digital-twin integration, which connects sensing outputs to project representations that support interpretation and visualization. Turkan et al. already demonstrated the value of linking scans to 4D information [1], while Kavaliauskas et al. integrated point clouds with IFC-based BIM to improve as-built object detection under clean and occluded conditions [5]. Vassena et al. further showed how repeated surveys from SLAM-based mapping devices can be compared with a 4D BIM over time to assess as-built versus as-planned conditions [6]. From a broader systems perspective, Pal et al. position such sensing-model integration within digital-twin architectures and highlight continuing methodological and technological barriers to predictive implementation [7]. This stream is highly relevant because it creates the technical basis for aligning field evidence with schedule representations, but it does not by itself define how those measurements should be converted into decision-relevant schedule updates.

The third stream is model-based schedule/progress updating. Hamledari et al. proposed an automated method to update IFC-based 4D BIMs by modifying schedule hierarchy and progress ratios within a unified environment [8]. This work is important because it moves beyond visualization toward formal updating of schedule-linked models. At the same time, much of this literature remains focused on updating element states or progress ratios inside the model environment, whereas management decisions still require an explicit translation from measured work into backlog, delay, forecast finish, and recovery targets.

Positioning the present paper against these three streams clarifies the research gap. The paper does not seek to contribute a new sensing algorithm, a new BIM platform, or a comprehensive schedule-forecasting benchmark. Rather, it contributes a methodological bridge between sensed work progress and schedule control: schedules and WBS structures are rarely authored with explicit attention to sensor observability and evidence requirements, and generalizable procedures for translating measured work into backlog, delay, forecast, and intervention targets remain under-specified. The framework proposed here addresses those two gaps while remaining compatible with existing automated monitoring, 4D BIM, and digital-twin pipelines.

3 Progress Monitoring versus Schedule Control

Progress monitoring and schedule control are often treated as interchangeable, but they answer different questions and operate on different variables. Progress monitoring is *production-facing*: It asks what work has been executed, at what rate, and how much remains. Schedule control is *management-facing*: It asks what the observed progress implies for activity finish dates, i.e. downstream impacts through dependencies, and what interventions are required to meet project objectives.

This separation aligns with established production-control thinking in construction. The Last Planner System popularized in Lean construction emphasizes production management to realize plans, supplementing traditional project control that often detects variances only after the fact [9]. Sensing and automation can greatly improve observability of executed work, but they do not eliminate the need for an explicit translation from work-domain measurements to time-domain schedule implications.

3.1 Different Units and Different Meanings

The distinction is reinforced by the use of different measurement units. Production monitoring is naturally expressed in work quantities and productivity (e.g. installed counts, length/area/volume/weight), while schedule control is expressed in time (start/finish dates, remaining durations, buffers, and floats). Percent complete can bridge these disparate measurement units only if it is anchored to a well-defined work quantity and a defensible evidence basis.

3.2 Evidence-Driven Completion

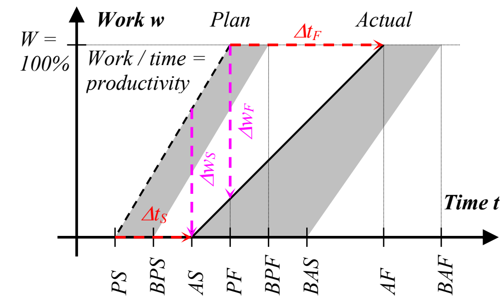
A central reason sensing does not automatically 'update the schedule' is that completion is not purely visual. Many activities have both an existence condition and a performance/acceptance condition. Existence is often verifiable through geometry or visual observation (sometimes for a limited time window before it becomes hidden, e.g. within walls). Performance and acceptance commonly require tests, inspections, certifications, commissioning, or approvals. When schedules conflate these notions, they either overstate what sensing can confirm or under-represent the non-visual gates that actually control formal status and payments.

3.3 Backlog And Delay As Translation Layer

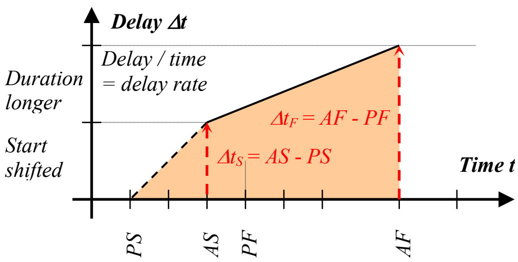
To connect work-based monitoring to time-based control, this paper uses two coupled indicators. Work backlog (or forelog) captures the shortfall (or surplus) of executed work relative to plan at a review time. Time delay expresses the time deviation implied by the achieved work relative to the planned production curve. Backlog is intuitive for field teams managing ongoing workflow and productivity, whereas delay and forecasts are necessary for schedule control, buffer management, and communication of downstream impacts. The next sections formalize these indicators and show how they support schedule updates and intervention targets.

Figure 1 summarizes the relationship between planned (P) and measured actual (A) performance to provide a common interpretation of delay and backlog. Its buffer (B) transmits impacts downstream to further activities. In the work-time view (Fig. 1a), the slope of each activity line represents productivity (work per time). Important dates on are its planned and actual starts (PS, AS) and planned and actual finishes (PF, AF). For generalizability, the activity produces a total of 100% at its completion, yet in practice any suitable unit of work quantity such as length, area, volume, weight, or count can be used

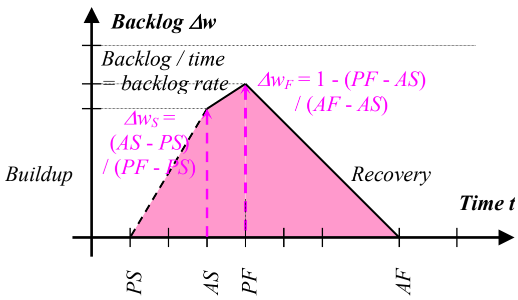
At any review time, including its start and finish (S, F), the vertical gap between planned and actual lines corresponds to work backlog Δw (or forelog if the actual is above plan), while the horizontal gap corresponds to time delay Δt. This information can more clearly be expressed in form of delay record (Fig. 1b) or work backlog that profiles build up and recovery (Fig. 1c).



1a: Linear Schedule



1b: Delay Record



1c: Work Backlog

Figure 1: Planned and actual activity performance (linear schedule, delay record, work backlog).

In Fig. 1b, the orange area marks the delay that accumulates, first at a steep delay rate of 1 delay-day for 1 time-day before the activity starts, then at a shallower rate when the activity progresses, albeit at a slower pace. Its shape is determined by the red arrows that mark the start delay time Δt_S, which is the difference of actual and planned starts, and the finish delay time Δt_F, which analogously is the difference between actual and planned finishes. Note that here the activity encounters both a start delay and a duration delay. While time always moves forward, delays that have accumulated can be recovered by an acceleration in the actual performance. In Fig. 1b, the purple area marks the work backlog that accumulates (at the same rate as the planned productivity in Fig. 1a) before the activity actually starts, then continues to grow at a lesser rate, because the actual productivity is lower than planned, and finally is diminished after the planned finish. Its shape is determined by the purple arrows at the actual start and planned finish. The actual start work backlog Δw_S is the difference of actual and planned starts (i.e. start delay) divided by the difference of planned finish and start (i.e. planned duration). And the planned finish backlog Δw_F is one minus a ratio that consists of planned finish minus actual start (i.e. overlap) divided by actual finish minus start (i.e actual duration).

4 Sensor-Aware Activity and WBS Design

Most progress monitoring pipelines presume that the project's WBS and schedule are already compatible with sensing-derived evidence. In practice, schedules are authored early and then optimized for coordination, contracting, and reporting, rather than for observability. A new sensor-aware approach is therefore needed at planning time to avoid 'orphan' activities whose fate cannot be ascertained by reliable evidence streams.

4.1 Deliverables, Evidence, and Observability

A WBS is commonly defined as a deliverable-oriented hierarchical decomposition of the entire work required to produce project deliverables [10]. This definition is useful for sensing integration, because sensors do not observe abstract activities; they record visible evidence about deliverables (installed objects, changed geometry, or resource presence). Sensor-aware authoring adds two attributes to each activity or work package: (i) Evidence type and (ii) evidence window.

Evidence type specifies what information sources can support progress and completion assessment: Visual/geometry (images, point clouds), resource-based sensing (positioning), documentary evidence (tickets, submittals, logs), quality assurance (QA) tests, or inspection/acceptance sign-offs. *Evidence window* specifies when that evidence exists and is observable (e.g. reinforcement visible before formwork closure; utilities visible before backfill). These attributes clarify what can be inferred from sensing and what must be obtained from QA and managerial processes.

4.2 Visual Observability across WBS Divisions

To illustrate why a sensor-aware WBS must distinguish between existence evidence and performance/acceptance evidence, a heuristic classification of common work divisions at the division level in the MasterFormat standard is carried out by assessing whether their progress can be determined visually from geometry or imagery (Figure 2).

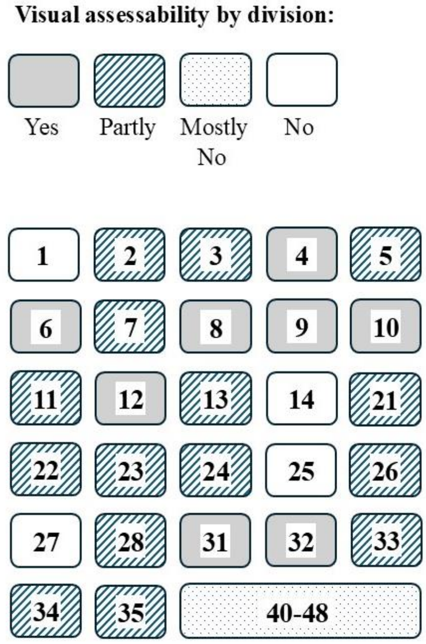


Figure 2: MasterFormat division classification of visual observability for progress monitoring.

Most divisions are only partly observable: Physical presence of installed work may be visible, but key acceptance conditions (e.g. compaction, weld quality, pressure rating, commissioning) require standardized tests, measurements, or formal documentation. This reinforces the need to represent non-visual gates explicitly in the schedule, rather than implicitly assuming that visible installation implies completion.

4.3 Completion Circuit with Explicit Gates

A naive response to partial observability is to split activities into many sub-activities, until each is measurable, often producing brittle schedules. Instead, this paper proposes a lightweight completion circuit that keeps schedules compact, while making non-visual gates explicit: Plan → approve → procure → execute → test/verify → accept. For many activities, execution remains the primary duration task supported by sensing-derived measurements, while test/verify and accept are represented as explicit milestones tied to QA documentation and approvals. This supports consistent schedule status updates without assuming that installed geometry alone implies acceptance.

A practical sensor-aware authoring workflow thus can be summarized as follows:

- Define deliverable scope for each work package and select an appropriate work unit (count, length, area, volume, weight; percent only if anchored to explicit work quantities).
- Assign evidence sources for execution progress (e.g. scans/images aligned to BIM) and for acceptance (tests, inspections, approvals).
- Specify observability windows and required sampling frequency, especially for work that becomes permanently concealed.
- Represent acceptance gates explicitly as milestones when they control formal status, even if no new geometry appears.

These steps are intentionally lightweight and are compatible with 4D BIM and digital twin implementations. They create the preconditions for translating measured work into time-based schedule updates and decision targets.

5 Backlog-Based Translation from Measured Work to Schedule Updates

This section defines a computable translation layer that converts measured work quantities into schedule-relevant indicators and intervention targets. The framework is agnostic to the sensing method; it requires only a defensible quantitative estimate of executed work at the chosen resolution, which can be produced by activity-level vision approaches aligned to 4D BIM [2], IFC-point cloud comparison methods [5], or repeated 4D BIM-SLAM surveys [6].

5.1 Cyclic Sensing-to-Scheduling Workflow

The sensing-to-scheduling concept can be viewed as a cyclic workflow that connects traditional 'bookends' of project control. The first bookend is the planning of activities and their measurement resolution, so that progress is supported by defensible evidence streams. The second bookend is the translation of measured work status into schedule updates and managerial actions (e.g. resource changes, re-sequencing, workface preparation) to recover performance. Tables 1 and 2 list these steps in chronological order, noting where sensing-derived measurements enter the loop and where backlog/delay calculations support schedule control decisions.

For activity i, let $W_{plan}(T)$ denote the planned cumulative work by time T, and $W_{actual}(T)$ the measured cumulative executed work. Planned work can be represented as a function of time (e.g. linear for constant planned productivity), while measured work can be obtained from sensing, logs, or hybrid data sources.

Table 1: Planning activities and measurement resolution

Phase	Action	Process
Activities	Define resolution	WBS
Estimating	Quantity Take-Off	Work count × dimensions = work quantity: $\#W \cdot l \cdot h \cdot d = W_{\text{plan}}$
Planning	Allocate Resources	Individual productivity: $\#R \cdot P_{\text{resource}} = P_{\text{plan}}$
		Work quantity / productivity = expected duration: $W_{\text{plan}} / P_{\text{plan}} = D_{\text{expected}}$
	Analyze risks	Risk factor × expected duration = planned duration: $R \cdot D_{\text{expected}} = D_{\text{plan}}$ $S_{\text{plan}} + D_{\text{plan}} = F_{\text{plan}}$
Scheduling	Sequence	Sequential or parallel activities: $w(t)_{\text{activity}} = \Sigma[w_{\text{i}}/d_{\text{i}}] \cdot \langle t - \Sigma[\text{d}_{i\text{-}1}]\rangle^1$
	Buffer	Continuous buffer after activity: $w(t)_{buffer} = w(t + b)_{\text{activity}}$
	Timing	Stack: Tentative successor start = max $[t(w)_{\text{buffer predecessor}}]$ Compact activities with buffers: $t(w)_{\text{successor}} = t(w)_{\text{successor tentative}}$ - min $[t(w)_{\text{successor tentative}} - t(w)_{\text{buffer predecessor}}]$
Controlling	Measure	Actual work quantity at time T: W_{actual}, best multiple updates
	Compare	Backlog at time T: $\Delta W(T) = w(t)_{\text{activity}} - W_{\text{actual}}$
	Compare	Delay at time T: $\Delta W(T) = T - t(W_{actual})_{\text{plan}}$
Visualizing	Plot plan	Work axis over time axis: $w_{\text{plan}}(t)$
	Plot actual	Work axis over time axis: $w_{\text{actual}}(t)$
	Plot delay	Difference plotted over time: $\Delta t(\text{w}) = t(w)_{\text{actual}} - t(w)_{\text{planned}}$
	Plot backlog	Difference plotted over time: $\Delta w(t) = w(t)_{\text{planned}} - w(t)_{\text{actual}}$

Table 2: Translating work status into schedule updates

Phase	Action	Process
Managing	Adjust	Accelerate or other decision to meet plan: $\#R_{\text{adjusted}} \cdot P_{\text{resource}} = P_{\text{adjusted}}$ $D_{\text{remaining}} = (W_{\text{plan}} - W_{\text{actual}}) / P_{\text{adjusted}}$
Scheduling	Time	Modify activity and buffer: $w(t)_{\text{adjusted}} = w(t < T)_{\text{plan}} + w(t \geq T)_{\text{remaining}}$
Visualizing	Adjust plan	Remaining progress: $F_{\text{adjusted}} = T + D_{\text{remaining}}$
	Adjust delay	Difference plotted over time: $\Delta t(w)_{\text{adjusted}} = t(w)_{\text{actual}} - t(w)_{\text{adjusted}}$
	Adjust backlog	Difference plotted over time: $\Delta w(t)_{\text{adjusted}} = w(t)_{\text{adjusted}} - w(t)_{\text{actual}}$

5.2 Backlog and Delay

Work backlog (positive) or forelog (negative) at review time T is defined as

$$\Delta W(T) = w(T)_{\text{activity}} - W_{\text{actual}}(T)$$

To express the same situation in time terms, define $t(w)_{plan}$ as the planned time when cumulative work w should have been achieved. Then the time delay is

$$\Delta T(T) = T - t(W_{actual})_{\text{plan}}$$

Backlog is a work-domain signal suitable for production control; delay is a time-domain signal required for schedule control and communication of downstream impacts.

5.3 Forecast and Recovery Targets

Using an estimated productivity P_{adjusted}, the remaining duration is

$$D_{\text{remaining}} = (W_{\text{plan}} - W_{\text{actual}}) / P_{\text{adjusted}}$$

and the forecasted finish is

$$F_{\text{adjusted}} = t + D_{\text{remaining}}$$

These computations turn measured progress into schedule updates for explicit intervention targets (e.g. resource changes or re-sequencing to increase productivity).

Execution forecasts should be paired with explicit acceptance gates from the sensor-aware authoring step. When acceptance depends on QA tests or approvals, those milestones remain separate schedule events whose dates may depend on additional evidence streams. The translation layer therefore supports realistic schedule updating without assuming sensor-only certification of completion.

6 Illustrative Example

This illustrative example is intentionally scoped as a methodological demonstration rather than a full empirical validation. Its purpose is to show the computability of the translation layer and the managerial interpretation of backlog, delay, forecast finish, and recovery targets while keeping acceptance explicit. The example therefore uses a single concrete-wall activity to demonstrate how sensing-derived work quantities can be converted into decision-relevant schedule information within the scope of a conference paper. Broader validation and comparison across multiple projects and across activity classes - including highly visible, partly visible, and QA-dominant work - are important next steps for future research.

6.1 Measured Progress at Midpoint

At the end of Day 5 (T = 5), sensing and model alignment indicate that $w_{\text{actual}}(5)$ = 50 m³ of wall concrete has been placed. Under the planned rate,

$$W_{\text{plan}}(5) = 12 \times 5 = 60 \text{ m}^3.$$

The backlog is therefore

$$\Delta W(5) = 60 - 50 = 10 \text{ m}^3.$$

6.2 Delay, Forecast, and Recovery Target

The planned time to reach 50 m³ is

$$t_{plan}(50) = 50/12 = 4.17 \text{ days},$$

implying a delay of

$$\Delta T(5) = 5 - 4.17 = 0.83 \text{ days}.$$

Remaining work is 120 - 50 = 70 m³. If productivity remains at 12 m³/day, remaining duration is

$$D_{\text{remaining}} = (120 - 50)/12 = 5.83 \text{ days}$$

and the forecast execution finish F_{adjusted} is 10.83, about 0.83 days late.

To still finish execution on Day 10, required productivity from Day 6 onward is 70 / (10 - 5) = 14 m³/day, an about 17% increase. This recovery target can be translated into feasible actions (e.g., additional crew, extended shifts, improved workface readiness, or resequencing). Importantly, the acceptance milestone remains distinct: even if sensing confirms existence of the placed wall, formal completion may still depend on test reports and inspections. The framework therefore provides decision-relevant schedule updates without over-claiming what sensing can certify.

7 Discussion

The proposed method is designed to complement progress-monitoring pipelines by making schedule-control transformations explicit. Its main value is in producing interpretable, computable indicators that connect measured work to time-based schedule consequences and intervention targets.

7.1 Applicability and Data Requirements

The framework is most effective when activities can be expressed in meaningful work units (count, length, area, volume) and when executed work can be measured frequently enough to support short-interval control. Repetitive and spatially distributed work, commonly monitored through periodic reality capture, is a good fit [1,6]. Hybrid settings are also viable when geometric measures are complemented with resource-based sensing or documentary evidence.

7.2 Limits: Hidden Work, Acceptance, and Uncertainty

The method does not eliminate fundamental limits of observability. Hidden work and short evidence windows can cause bias if capture frequency is misaligned with construction operations, echoing prior findings that planning for scanning affects monitoring performance [1]. In practical implementations, sensing outputs may be affected by occlusion, classification error, incomplete scans, and limited evidence windows in which work is observable before it becomes concealed. Acceptance often depends on QA tests, inspections, and approvals that are not inferable from geometry; these must remain explicit milestones in the schedule. Backlog and delay therefore remain useful decision signals, but when they are used to support risk assessment or float and buffer decisions, they should ultimately be paired with uncertainty bounds rather than interpreted as exact values. The present paper focuses on establishing the core deterministic translation framework; extending it with probabilistic uncertainty modeling is therefore an important direction for future research.

7.3 Integration with 4D BIM and Digital Twins

The method can be implemented as a management layer within 4D BIM and digital twin architectures: sensing provides $W_{\text{actual}}(T)$, the translation layer computes backlog, delay, and forecasts, and the schedule representation stores updated statuses and explicit gate milestones. Digital twin reviews highlight that practical implementation still faces methodological gaps and integration challenges [7]; a clear translation layer and evidence-aware activity definitions provide a concrete step toward closing those gaps. IFC-based schedule updating workflows can also benefit by pairing progress ratio updates with backlog/delay indicators that make time implications explicit [8].

8 Conclusions and Future Research

Automated sensing and progress estimation are advancing quickly, but project control still requires a defensible bridge from measured work to schedule updates and managerial actions. This paper contributes a management-oriented sensing-to-scheduling method that remains realistic about what sensing can verify.

The key contributions are: (i) An explicit separation between production-facing progress monitoring and schedule control; (ii) sensor-aware activity and WBS authoring that distinguishes existence evidence from performance/acceptance evidence and represents acceptance gates explicitly; and (iii) a backlog-based translation layer that converts measured work into backlog/forelog, delay, forecast finishes, and recovery targets suitable for schedule updating.

Because the present paper is scoped as a methodological conference contribution, future research should validate and compare the framework across diverse project datasets and activity classes (highly visible work, partly visible work, and QA-dominant work). It should also fuse sensing-derived existence measures with QA/test and inspection systems to automate gate milestones where appropriate and represent measurement uncertainty explicitly through range forecasts and risk-aware interpretations of float and buffer consumption. Empirical evaluation of how backlog/delay indicators influence decision timing and mitigation effectiveness is also needed to quantify benefits relative to percent-complete-only reporting.

9 References

[1] Y. Turkan, F. Bosché, C. Haas, and R. Haas. Automated progress tracking using 4D schedule and 3D sensing technologies. Automation in Construction, 22:414–421, 2012.

[2] A. Pal, J. J. Lin, S.-H. Hsieh, and M. Golparvar-Fard. Activity-level construction progress monitoring through semantic segmentation of 3D-informed orthographic images. Automation in Construction, 157:105157, 2024.

[3] M. S. U. Rehman, M. T. Shafiq, and F. Ullah. Automated computer vision-based construction progress monitoring: A systematic review. Buildings, 12(7):1037, 2022.

[4] J. Zhao, E. Pikas, O. Seppänen, and A. Peltokorpi. Using real-time indoor resource positioning to track the progress of tasks in construction sites. Frontiers in Built Environment, 7:661166, 2021.

[5] P. Kavaliauskas, J. B. Fernandez, K. McGuinness, and A. Jurelionis. Automation of construction progress monitoring by integrating 3D point cloud data with an IFC-based BIM model. Buildings, 12(10):1754, 2022.

[6] G. P. M. Vassena, L. Perfetti, S. Comai, S. Mastrolembo Ventura, and A. L. C. Ciribini. Construction progress monitoring through the integration of 4D BIM and SLAM-based mapping devices. Buildings, 13(10):2488, 2023.

[7] A. Pal, J. J. Lin, S.-H. Hsieh, and M. Golparvar-Fard. Automated vision-based construction progress monitoring in built environment through digital twin. Developments in the Built Environment, 16:100247, 2023.

[8] H. Hamledari, B. McCabe, S. Davari, and A. Shahi. Automated schedule and progress updating of IFC-based 4D BIMs. Journal of Computing in Civil Engineering, 31(4):04017012, 2017.

[9] H. G. Ballard. The Last Planner System of Production Control. PhD dissertation, University of Birmingham, 2000.

[10] Project Management Institute. Practice Standard for Work Breakdown Structures, Second Edition. Project Management Institute, Newtown Square, PA, 2006.