# Reinforcement Learning for Adaptive Control of Robotic Construction Processes Under Variable Material Conditions

Niclas Said
School of Computing, Clemson University, Clemson, SC, USA.
niclassaid679@clemson.edu

**Abstract**

Robotic construction processes operate under material conditions that shift with mixing, curing, moisture, temperature, delivery pressure, and layer geometry. A control policy tuned for a nominal mortar or concrete rheology can lose deposition accuracy or violate process limits when yield stress, stiffness, and flow resistance change during execution. This manuscript develops a methodological framework for adaptive control of robotic construction processes by coupling online material-state estimation with safety-constrained reinforcement learning. The robot is modeled as acting on a belief over latent material and process parameters rather than on a single nominal state. A recursive estimator updates those parameters from force, torque, pressure, vision, and kinematic residuals, while the policy learns continuous tool-motion and deposition commands under reward terms for geometric quality, productivity, actuator effort, and constraint violation. Safety is represented through constrained policy optimization with explicit limits on layer stability, collision margin, motor torque, and surface tolerance. A sim-to-real discrepancy bound is used to structure calibration and domain randomization, and uncertainty-sensitive action selection is formulated to avoid aggressive adaptation when material identification is weak. The framework clarifies how estimator covariance, constraint thresholds, reward weights, and simulation fidelity jointly determine adaptation behavior. Its central contribution is an integrated control logic that treats material variability as a partially observable, safety-relevant state rather than as an exogenous disturbance. The analysis provides design criteria for robotic construction systems that must remain productive without sacrificing process stability.

**Keywords**

reinforcement learning; adaptive control; robotic construction; material variability; constrained policy optimization; sim-to-real transfer

## 1. Introduction

Reinforcement learning has become a practical route for acquiring closed-loop robot control policies in tasks where accurate analytical models are difficult to obtain [1]. The standard formulation treats control as a Markov decision process in which an agent selects actions to maximize cumulative reward under uncertain transitions [2]. Continuous-control algorithms have extended this formulation to high-dimensional actuation through deterministic policy gradients [3], while policy-gradient methods such as proximal policy optimization provide stable updates for stochastic policies [4]. End-to-end visuomotor learning has further shown that perception and action can be coupled when demonstrations or interaction data are available [5]. These developments are relevant to construction robotics

because construction tasks combine noisy sensing, continuous actuation, contact-rich manipulation, and process constraints that are difficult to encode as fixed scripts.

Construction automation has advanced through task-specific machines, programmable manipulators, and integrated building-production systems [6]. Robotic fabrication research has examined dynamic slip-form casting and eco-friendly façade design, illustrating how tool motion and material delivery must be coordinated with geometric and material requirements [7]. The unresolved control problem is not simply one of path following. Construction materials are variable by nature: fresh concrete can change yield stress and viscosity during transport, extrusion pressure can fluctuate with nozzle geometry, and ambient conditions alter curing and layer stability. A policy optimized for a nominal material may deposit too little or too much material, induce excessive contact force, or create local geometry errors that propagate into later layers. Fixed-parameter control therefore has limited robustness when material state is spatially heterogeneous and temporally nonstationary.

The research question addressed here is how a robotic construction process can adapt its control policy to latent and changing material conditions while respecting safety and quality constraints. The proposed answer is a reinforcement-learning framework in which the policy acts on a probabilistic belief over material and process parameters. The framework combines recursive estimation, constrained policy optimization, uncertainty-sensitive action selection, and a sim-to-real calibration logic. Its central contribution is not a new benchmark result but an integrated control formulation that specifies what must be estimated, how uncertainty enters action selection, how construction constraints enter policy learning, and how simulation fidelity affects transfer. The framework is developed analytically so that design choices can be assessed before they are embedded in a particular robot or material system.

## 2. Problem Formulation and Material-Condition State Representation

The construction process is modeled as a partially observable Markov decision process because the robot cannot directly measure all material properties that affect deposition [8]. The observable state includes end-effector pose and velocity, tool reaction forces, extrusion pressure or delivery rate, layer height, surface profile, and actuator temperatures. The action includes joint velocity or torque commands, tool travel speed, nozzle orientation, and material flow commands. The latent material state includes yield stress, apparent viscosity, curing rate, moisture content, aggregate distribution, and compactability. Probabilistic state estimation provides the general logic for maintaining a belief over such hidden variables from noisy measurements [9].

A compact transition model can be written as

$$x_{t+1} = f(x_t, a_t, \theta_t) + w_t$$

where $x_t$ is the observable process state, $a_t$ is the control action, $\theta_t$ is the vector of latent material and process parameters, $w_t$ is process noise, and f represents the coupled robot-material dynamics. The model is not assumed to be known perfectly; it represents the causal structure that the estimator and policy must accommodate. The material vector $\theta_t$ matters because two states that look identical in pose and velocity can require different flow or force commands when yield stress and stiffness differ.

Material variability in construction is well documented in extrusion and casting research. Contour crafting and related automated construction methods depend on coordinated material delivery and tool motion [10]. Extrusion-based concrete printing requires control of

pumpability, buildability, and open time, all of which affect the feasible process window [11]. Structural build-up of cement-based materials further shows that layer stability depends on time-dependent rheology rather than on a fixed material constant [12]. These considerations imply that the controller must update its material estimate during execution and must treat the estimate as uncertain.

A recursive material-parameter update can be expressed as

$$\hat{\theta}_{t+1} = \hat{\theta}_t + K_t(y_t - h(x_t, \hat{\theta}_t))$$

where $\hat{\theta}_t$ is the current estimate, $y_t$ is the measurement vector, h is the measurement model, and $K_t$ is an estimator gain. The innovation $y_t - h(x_t, \hat{\theta}_t)$ captures the mismatch between predicted and observed force, pressure, or geometry. The gain can be reduced when the estimated covariance is large or when measurements are noisy, preventing a single anomalous reading from causing an aggressive policy change. The estimator also maintains a covariance $\Sigma_t$ that represents uncertainty in $\theta$; this covariance is used later in action selection. The update is intentionally generic so that it can be instantiated as an extended Kalman filter, a particle filter, or a residual-learning estimator depending on sensor suite and process dynamics.

## 3. Reinforcement Learning Control Architecture

The control policy maps the current belief to a continuous action. Let $b_t$ denote the belief state constructed from $x_t$, $\hat{\theta}_t$, and $\Sigma_t$. The policy objective is

$$J(\pi) = E[\sum_t \gamma^t r(x_t, a_t, \theta_t)]$$

where $\pi$ is the policy, $\gamma$ is a discount factor, and r is the stage reward. The expectation is taken over process noise, material uncertainty, and policy stochasticity. This objective is standard in reinforcement learning [2], but its construction differs from nominal robot control because the reward depends on latent material parameters and because the policy must remain safe while those parameters are being identified.

Classical adaptive control provides a useful reference point. Adaptive laws adjust controller parameters in response to tracking error and parameter drift [13]. The present framework retains that emphasis on online adjustment but replaces hand-tuned adaptation laws with a learned policy that can represent nonlinear interactions among tool velocity, flow command, contact force, and layer geometry. Model-based probabilistic policy search offers a complementary route when data are scarce, because it can exploit approximate dynamics and uncertainty estimates during learning [14]. A hybrid design is therefore plausible: a probabilistic model supports short-horizon prediction and safety checks, while a learned policy supplies the closed-loop action.

The architecture separates four functions. Perception produces state and geometry estimates. Material identification maintains $\hat{\theta}_t$ and $\Sigma_t$. The policy selects actions from the belief. A safety layer filters or projects actions that would violate hard process limits. This separation is important because material estimation errors and policy errors have different consequences. A slow material estimate may reduce productivity, whereas an unconstrained policy action may cause collision, motor overload, or layer collapse. The learning problem should therefore optimize performance under constraints rather than treat safety as a post hoc penalty only.

## 4. Adaptive Estimation and Uncertainty-Aware Policy Learning

Material identification is most reliable when the robot excites the process in a controlled way. A deposition pass that varies speed and flow modestly can reveal flow resistance and stiffness, but excessive excitation can damage the layer or produce unacceptable geometry. The controller therefore needs an action-selection rule that balances expected reward against uncertainty. A practical form is

$$a_t = argmax_a \left[ Q(b_t, a) - \kappa \sqrt{\sigma^2(b_t, a)} \right]$$

where $Q(b_t, a)$ is the expected return from belief $b_t$ and action a, $\sigma^2(b_t, a)$ is the predictive variance associated with that action, and $\kappa$ is a nonnegative uncertainty weight. When $\kappa$ is small, the controller exploits the current estimate aggressively. When $\kappa$ is large, the controller becomes conservative in regions where material parameters are poorly identified. The variance term can be derived from $\Sigma_t$ and from learned heteroscedastic uncertainty in the value model. This rule does not guarantee safety by itself; it reduces the chance that high-reward but poorly understood actions are selected without scrutiny.

The estimator and policy should operate on different time scales. Material parameters such as yield stress may change over seconds to minutes, whereas joint-level servo control occurs at milliseconds. A hierarchical design is therefore appropriate: a fast feedback loop stabilizes tool motion and contact force, and a slower adaptation loop updates material belief and policy context. The adaptation gain can be scheduled as a function of innovation magnitude and covariance, so that large persistent residuals trigger learning while isolated outliers are down-weighted. This is consistent with adaptive-control practice, where adaptation must be bounded to avoid instability under noise and unmodeled dynamics [13].

Exploration must also be constrained by process physics. In construction, an exploratory action is not merely a low-reward action; it can produce a defective layer that invalidates subsequent work. The policy should therefore explore within admissible ranges of flow, speed, and force, and the safety layer should reject actions whose predicted consequences exceed layer-stability or collision margins. The resulting learning problem is closer to constrained reinforcement learning than to unconstrained reward maximization. This distinction becomes explicit in the next section.

## 5. Safety-Constrained Optimization and Reward Design

Construction constraints are multi-objective and partly hard. Geometric tolerance and collision avoidance are hard limits. Layer stability, motor torque, and surface finish may be soft during early learning but hard during execution. Constrained policy optimization provides a formal way to update policies while approximating constraint satisfaction [15]. Constrained Markov decision process theory supplies the underlying formulation in which reward maximization is subject to expected cost bounds [16]. Safe reinforcement learning surveys further emphasize that constraint handling must be designed into the learning problem rather than added after training [17].

A stage reward suitable for robotic deposition can be written as

$$r_t = r_prod - \lambda_m \, e_m^2 - \lambda_s \, C_s - \lambda_a \| a_t - a_{t-1} \|^2$$

where r_prod rewards completed productive deposition, $e_m$ is a geometric or material-placement error, $C_s$ is a nonnegative safety-cost term, and $\lambda_m$, $\lambda_s$, and $\lambda_a$ are nonnegative weights. The term $\|a_t - a_{t-1}\|^2$ penalizes chattering and abrupt changes that can destabilize

material flow. The reward is intentionally decomposable: geometric error, safety cost, and actuator smoothness can be inspected separately when tuning behavior. The safety cost may aggregate collision margin violation, torque saturation, excessive layer deflection, or pressure outside the admissible range.

The policy optimization problem can be stated as

where $R(\pi)$ is cumulative reward, $C_i(\pi)$ is the i-th expected cost, and $d_i$ is its admissible threshold. The thresholds should be derived from design specifications, material limits, and actuator ratings rather than from arbitrary preferences. This formulation clarifies why simply increasing a penalty weight is insufficient: a penalty changes the trade-off, whereas a constraint defines an admissible region. During execution, a safety shield can still project actions that violate predicted hard limits, because even a well-trained constrained policy may encounter out-of-distribution material conditions.

The choice of constraint thresholds determines the operating envelope. Tight thresholds increase safety but may force the robot to slow down or request human intervention when material uncertainty is high. Loose thresholds increase productivity but reduce the margin against layer collapse or geometric drift. The framework therefore treats constraint design as part of control design, not as a separate compliance step. Material estimation uncertainty should be propagated into the predicted cost, so that an action with acceptable mean cost but high variance is treated cautiously.

**6. Sim-to-Real Transfer and Calibration Logic**

Learning directly on a construction site is expensive and risky, so simulation and calibrated models are necessary. Deep reinforcement learning practice shows that transfer depends strongly on observation design, reward specification, dynamics randomization, and systematic evaluation [18]. Domain randomization can improve transfer by exposing the policy to varied visual and physical conditions during training [19]. Dynamics randomization extends this idea to mass, friction, actuation delay, and other physical parameters, allowing the policy to learn robust behavior across a distribution of environments [20]. In construction, the randomized variables should include material rheology, delivery pressure, nozzle wear, sensor noise, and layer geometry.

A sim-to-real discrepancy bound provides a design-oriented way to reason about transfer. For a fixed policy $\pi$, a simplified bound can be written as

$$|J_real(\pi) - J_sim(\pi)| \leq L_f\, E[\|\Delta\xi\|] + L_r\, E[\|\Delta r\|]$$

where J_real and J_sim are real and simulated returns, $\Delta\xi$ represents model-parameter discrepancy, $\Delta r$ represents reward or measurement discrepancy, and L_f and L_r are Lipschitz-style sensitivity constants. The bound is not a numerical guarantee for a particular site; it expresses the conditions under which transfer is expected to degrade. Reducing discrepancy in dynamics and reward, and reducing the policy's sensitivity to those discrepancies, both improve the expected transfer. This justifies calibration of material parameters, reward normalization, and conservative uncertainty penalties during early deployment.

Calibration logic should combine randomized simulation with online system identification. Before execution, priors for yield stress, viscosity, and flow resistance can be initialized from material tests and prior batches. During execution, residuals between predicted and measured force, pressure, and geometry update the belief. If residuals remain

large, the controller should reduce speed or flow, increase sensing frequency, or request operator confirmation. The safety layer should use the calibrated model plus uncertainty bounds to reject actions whose predicted consequences are outside admissible limits. This logic does not eliminate the sim-to-real gap; it makes the gap observable and actionable.

## 7. Technical Discussion: Failure Modes, Sensitivity, and Trade-offs

The framework's main failure modes are identifiable. Material misidentification occurs when sensor residuals are ambiguous, for example when changes in flow resistance and nozzle clogging produce similar pressure signatures. A policy may then adapt in the wrong direction. This risk is reduced by maintaining covariance, using multiple sensing modalities, and limiting adaptation rate when innovation is poorly conditioned. Another failure mode is reward exploitation: a policy may achieve geometric reward by slowing deposition excessively or by making many small passes that satisfy local error but reduce productivity. The reward decomposition and constraint formulation limit this behavior, but weight selection remains consequential.

Nonstationary curing is a second stress case. A material that is stable at one layer height may become unstable as time passes, so the feasible action set shrinks during execution. The belief state must include time since deposition and ambient conditions, and the safety layer must update admissible flow and speed accordingly. If the policy treats curing as stationary, it may continue with a flow rate that was safe earlier. The constrained formulation helps only if the cost model reflects time-dependent stability; otherwise the constraint is mis-specified.

Sensitivity analysis should focus on the uncertainty weight $\kappa$, the constraint thresholds $d_i$, and the reward weights. Large $\kappa$ makes the controller conservative and may cause unnecessary slowdowns in well-understood regions. Small $\kappa$ encourages exploitation but can produce aggressive actions when material identification is weak. Tight constraint thresholds preserve safety but may make the learning problem infeasible under noisy sensing; loose thresholds admit unsafe behavior. Reward weights trade geometric accuracy against speed and smoothness. These are not merely tuning parameters; they encode the desired operating policy of the construction system.

The framework also involves trade-offs between model-based and model-free components. A purely model-free policy can learn complex behaviors but may require extensive interaction data and may transfer poorly when material statistics shift. A purely model-based controller can generalize with less data but depends on the fidelity of the process model. The proposed separation of estimation, policy, and safety layer allows each component to use the most appropriate representation. The policy can be learned, the estimator can remain probabilistic, and the safety layer can remain rule-based or optimization-based. This modularity is a strength, but it also creates interface risks: errors in belief construction propagate into policy decisions, and overly conservative safety filters can obscure learning signals.

## 8. Conclusion

This manuscript has developed a reinforcement-learning framework for adaptive control of robotic construction processes under variable material conditions. The framework treats material state as a partially observable, safety-relevant variable and couples recursive estimation with a policy that acts on a belief rather than a nominal model. Uncertainty enters action selection through a variance penalty, while geometric quality, productivity, actuator smoothness, and process safety enter through a decomposed reward and explicit constraints.

A sim-to-real discrepancy bound clarifies how model error, reward mismatch, and policy sensitivity affect transfer, and calibration logic connects randomized simulation with online material identification.

The central design insight is that adaptation in construction robotics cannot be reduced to faster parameter tuning. Material variability changes the feasible action set, the cost of exploration, and the reliability of sensing. A control policy must therefore reason about uncertainty and constraints simultaneously. The framework provides a coherent basis for specifying estimators, reward terms, constraint thresholds, and safety filters for robotic deposition and casting processes. Its practical value lies in making the interaction among material belief, policy adaptation, and process safety explicit, so that future implementations can be evaluated against clear design criteria rather than nominal-condition performance alone.

**References**


1. Kober, J., Bagnell, J. A., & Peters, J. (2013). Reinforcement learning in robotics: A survey. International Journal of Robotics Research, 32(11), 1238–1274.

2. Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction (2nd ed.). MIT Press.

3. Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., & Wierstra, D. (2016). Continuous control with deep reinforcement learning. In International Conference on Learning Representations.

4. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. arXiv:1707.06347.

5. Levine, S., Finn, C., Darrell, T., & Abbeel, P. (2016). End-to-end training of deep visuomotor policies. Journal of Machine Learning Research, 17(39), 1–40.

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

7. Yu, L., Luo, D., & Xu, W. (2018). Dynamic robotic slip-form casting and eco-friendly building façade design. In Robotic fabrication in architecture, Art and Design (pp. 421-433). Cham: Springer International Publishing.

8. Kaelbling, L. P., Littman, M. L., & Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. Artificial Intelligence, 101(1–2), 99–134.

9. Thrun, S., Burgard, W., & Fox, D. (2005). Probabilistic robotics. MIT Press.

10. Khoshnevis, B. (2004). Automated construction by contour crafting—Related robotics and information technologies. Automation in Construction, 13(1), 5–19.

11. Buswell, R. A., Leal de Silva, W. R., Jones, S. Z., & Dirrenberger, J. (2018). 3D printing using concrete extrusion: A roadmap for research. Cement and Concrete Research, 112, 58–69.

12. Perrot, A., Rangeard, D., & Pierre, A. (2016). Structural built-up of cement-based materials used for 3D-printing extrusion techniques. Materials and Structures, 49(4), 1213–1220.

13. Åström, K. J., & Wittenmark, B. (2008). Adaptive control (2nd ed.). Dover Publications.

14. Deisenroth, M. P., & Rasmussen, C. E. (2011). PILCO: A model-based and data-efficient approach to policy search. In Proceedings of the 28th International Conference on Machine Learning (pp. 465–472). Omnipress.

15. Achiam, J., Held, D., Tamar, A., & Abbeel, P. (2017). Constrained policy optimization. In Proceedings of the 34th International Conference on Machine Learning (pp. 22–31). PMLR.

16. Altman, E. (1999). Constrained Markov decision processes. Chapman & Hall/CRC.

17. Garcia, J., & Fernández, F. (2015). A comprehensive survey on safe reinforcement learning. Journal of Machine Learning Research, 16(1), 1437–1480.

18. Ibarz, J., Tan, J., Finn, C., Kalakrishnan, M., Pastor, P., & Levine, S. (2021). How to train your robot with deep reinforcement learning: Lessons we have learned. International Journal of Robotics Research, 40(4–5), 698–721.

19. Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., & Abbeel, P. (2017). Domain randomization for transferring deep neural networks from simulation to the real world. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 23–30). IEEE.

20. Peng, X. B., Andrychowicz, M., Zaremba, W., & Abbeel, P. (2018). Sim-to-real transfer of robotic control with dynamics randomization. In 2018 IEEE International Conference on Robotics and Automation (pp. 3803–3810). IEEE.