Generative Design Optimization of Modular Masonry Structures Using Computational Geometry and Robotic Fabrication Constraints

Edith Devaux
Department of Computer Science, Colorado State University, Fort Collins, CO, USA.
edevaux@colostate.edu

Sosimo Fuertes
Department of Computer Science, Binghamton University, Binghamton, NY, USA.
sosimo.fuertes@binghamton.edu

Abstract

Modular masonry structures combine discrete compression-resistant units with free-form architectural surfaces, but their design is constrained by joint stability, module geometry, construction sequence, and the reachable workspace of robotic manipulators. Existing digital workflows often separate geometric modeling, structural form finding, and fabrication planning, so a layout that is visually or structurally plausible may be difficult or impossible to assemble. This paper develops a generative design optimization framework that treats module position, orientation, unit type, and assembly order as coupled decision variables. A discrete limit-state formulation enforces compression-dominated equilibrium and Coulomb friction at joints; computational geometry relates surface curvature to admissible module size, joint deviation, and bond staggering; and robotic constraints encode inverse-kinematic reachability, collision clearance, payload limits, and placement tolerance. The resulting multiobjective problem minimizes material volume, fabrication time, surface deviation, and constraint penalties, while a robust margin measures feasibility under bounded variations in module dimensions, friction, and placement error. The framework is assessed analytically by tracing constraint interactions rather than through reported experiments. The analysis indicates that curvature-adaptive module sizing and sequence-aware reachability are first-order determinants of feasible designs, and that ignoring either produces layouts whose structural or fabrication feasibility is only incidental. The formulation provides a coherent basis for computational design tools that align geometric freedom with compression-only masonry behavior and robotic fabrication limits, reducing the need for late-stage redesign and supporting modular, robotically assembled masonry construction.

Keywords

modular masonry; generative design; computational geometry; robotic fabrication; limit analysis; multiobjective optimization

1. Introduction

Modular masonry is a construction logic in which repeated discrete units are assembled into a continuous structural surface. Its architectural appeal lies in the visible rhythm of the module, the material efficiency of compression-dominated load paths, and the possibility of prefabrication or robotic placement. Its computational difficulty lies in the coupling between geometry and buildability. A free-form surface must be decomposed into units that fit

together, transmit compression, resist sliding through friction, and remain reachable by a manipulator without collisions. Architectural geometry provides the mathematical vocabulary for such decompositions, including surfaces, meshes, offsets, and curvature-aware discretization [1]. Digital design research has long emphasized that geometric representation is not a neutral medium but a set of constraints and affordances that shape what can be designed and built [2]. When masonry is treated as a discrete system, those affordances become structural and robotic rather than merely visual.

Compression-only equilibrium is the central structural condition. Lower-bound limit analysis offers a practical route for assessing whether a stable force state exists inside a masonry assembly, and it has been used to analyze vaulted masonry buildings under realistic loading assumptions [3]. Form-finding methods such as particle-spring systems provide complementary tools for generating funicular geometries that are naturally suited to compression [4]. Computational masonry models have extended these ideas to discrete voussoirs and unreinforced assemblies, linking geometric design to stability and fabrication [5]. Assembly planning adds another layer: even a stable final configuration may be unreachable if modules cannot be inserted in a feasible sequence or if temporary supports cannot be removed [6].

Generative design methods are appropriate when the design space is combinatorial and the objectives conflict. Practical generative design frameworks combine parametric variation, constraint evaluation, and search rather than relying on a single closed-form solution [7]. In parallel, digital fabrication research has shown that material systems, tooling, and assembly processes should be treated as design variables rather than downstream constraints [8]. On-site automatic brick masonry systems indicate that robotic placement of modular units is technically feasible, yet their planning models generally treat geometry as an input rather than as an optimized variable [9]. The resulting gap is not a lack of individual techniques but a lack of a coupled formulation in which computational geometry, compression-only equilibrium, and robotic fabrication constraints are evaluated within the same generative search.

This paper addresses that gap by developing a generative design optimization framework for modular masonry structures under explicit computational-geometric and robotic-fabrication constraints. The central contribution is a coupled decision model in which module layout, unit selection, joint orientation, and assembly sequence are optimized against structural feasibility, surface fidelity, and manipulator reachability. The framework is methodological and analytical: it defines the decision variables, constraint functions, objective terms, and robust feasibility margins required for such optimization, and it examines the resulting design trade-offs without claiming experimental validation. The aim is not to replace detailed structural analysis or robot programming, but to provide a rigorous intermediate representation that prevents geometrically attractive but structurally or robotically infeasible layouts from entering detailed design.

2. Problem Formulation

2.1 Discrete module representation

The structure is represented as a finite set of masonry modules attached to a target surface or enclosed volume. Each module has a position, orientation, and type drawn from a catalog of admissible units. Let N be the number of modules and let the design vector be

$$X = \{m_i\}_{i=1}{}^{N}, m_i = (x_i, y_i, z_i, \theta_i, b_i)$$

where x_i, y_i, and z_i locate the module reference point, θ_i denotes its orientation about the surface normal or a local frame, and b_i indexes the unit type from a finite catalog. The number N is itself variable, so the formulation permits both denser layouts with small units and coarser layouts with large units. The variables are not independent: each module must fit its neighbors within joint tolerances, maintain a minimum bearing area, and preserve the intended bond pattern. These conditions define a feasible configuration space that is nonconvex and partly discrete.

The representation deliberately separates the geometric surface from the discrete module set. A continuous target surface can be tessellated into candidate module positions, but the final design is a discrete assembly. This separation is useful because structural equilibrium and robotic reachability are evaluated on the discrete assembly, while curvature and surface deviation are evaluated on the continuous target. The two representations are linked by a projection or fitting operator that maps each module to the target surface and computes local geometric error.

2.2 Structural feasibility

Structural feasibility is formulated as the existence of a compression-dominated equilibrium state. For a given layout X, let f be the vector of contact force resultants at joints and let g be the vector of applied loads. Static equilibrium requires

$$A(X)\, f \;=\; g, f \;\in\; K(X) \;=\; \{f_j\colon f_{nj} \;\geq\; 0, \|f_{tj}\| \;\leq\; \mu\, f_{nj} \;+\; c_j\}$$

where A(X) is the equilibrium matrix assembled from module geometry, f_{nj} and f_{tj} are the normal and tangential components of the joint force f_j, μ is the friction coefficient, and c_j is an optional cohesion term that is zero for dry-stacked joints and small for mortared joints. The set K(X) encodes the no-tension condition through $f_{nj} \geq 0$ and the friction cone through the tangential bound. The matrix A(X) depends on module size and orientation because changing the geometry changes lever arms, contact locations, and load paths.

This is a lower-bound limit-state condition rather than a full finite-element description. It asks whether at least one admissible force state can support the loads, which is the appropriate question during generative search because the final stress distribution in masonry is statically indeterminate and depends on construction history. The constraint is evaluated as a feasibility problem: if no f satisfies equilibrium and the friction cone, the layout is rejected or penalized. The formulation also permits a stability margin by requiring f_{nj} to exceed a minimum compressive prestress or by shrinking the admissible friction cone during optimization.

2.3 Robotic fabrication constraints

Robotic fabrication imposes kinematic and geometric constraints that are independent of structural feasibility. For each module i, let $T_i(X)$ be the target tool frame, let F(q) be the forward kinematic map of the manipulator, and let Q be the set of joint configurations satisfying joint limits and velocity bounds. The reachability residual is

$$q_i \;=\; argmin_\{q \in Q\}\, \|T_i(X) \;-\; F(q)\|_2, e_i \;=\; \|T_i(X) \;-\; F(q_i)\|_2 \;\leq\; \varepsilon$$

where ε is the admissible positioning error. A module is reachable only if the residual is below tolerance and the configuration q_i avoids singularities. Reachability depends on the robot base placement and on the workpiece position, so both can be treated as optimization variables or as scenario parameters.

Collision avoidance is evaluated against the partially built structure, temporary supports, and the robot itself. Let $R(q_i)$ be the set of points occupied by the robot at configuration q_i and let S be the obstacle set. The clearance condition is

$$d_i = min_\{p \in R(q_i), s \in S\} \|p - s\|_2 \geq \delta$$

where δ is the required clearance. For placement paths, q_i is replaced by a trajectory and the minimum is taken over all configurations along the path. Payload limits are imposed separately as $m_i g \leq P_{max}$, where m_i is module mass and P_{max} is the rated payload. Tool geometry, gripper width, and mortar application also restrict accessible orientations and insertion directions.

3. Computational Geometry of Modular Masonry

The geometric layer determines whether a target surface can be decomposed into modules that satisfy joint tolerances and bond requirements. Curvature is the primary constraint. On a smooth surface with principal curvatures κ_1 and κ_2, the angular deviation across a module of characteristic size h is approximately proportional to κh. If the permissible angular mismatch at a joint is $\delta\theta$, then the module size must satisfy

$$h_i \leq \delta \frac{\theta}{\max}(|\kappa_1|, |\kappa_2|)$$

where the maximum is taken over the local patch occupied by module i. The inequality makes explicit why high-curvature regions require smaller units or thicker joints: the same module size produces larger normal deviations and weaker bearing conditions. In practice, the bound is conservative because joints can accommodate some rotation, but it provides a principled initial filter before detailed contact analysis.

Seam orientation is equally important. Joints should be oriented to avoid continuous straight lines that can form slip planes, and bond patterns should stagger vertical joints between courses. On curved surfaces, staggering must be defined with respect to geodesic directions rather than global coordinates. A practical representation uses a parametric or mesh-based surface with a local orthonormal frame, computes a preferred joint direction from the principal curvature directions, and then alternates the bond offset by half a module between adjacent courses. The resulting adjacency graph records which modules share a bearing surface and which joints are potential slip planes. This graph is used by the structural feasibility check and by the assembly sequence planner.

Surface fidelity is measured as the integral distance between the target surface and the outer faces of the modules. A layout with many small modules can approximate the target closely but increases fabrication time and the number of joints. A layout with large modules reduces the number of placements but may exceed curvature tolerances or create open joints. The computational-geometric contribution is therefore not a single tessellation algorithm but a set of constraints that make the trade-off explicit and searchable. Prior discrete masonry models show that geometric decisions such as voussoir subdivision and joint layout directly affect stability [5], and assembly-planning work demonstrates that the order of placement can determine whether a stable structure is constructible [6].

4. Robotic Fabrication Constraints and Buildability

Robotic construction systems shift the feasibility boundary from geometric possibility to manipulator capability. Construction robotics research has established that task planning, tooling, and site logistics are tightly coupled in automated building [10]. Offsite and onsite

automation further require that module tolerances, handling, and sequencing be designed together rather than checked after fabrication [11]. Digital materiality research has similarly emphasized that the robot is not a neutral executor but a participant in the material and geometric logic of the assembly [12]. These perspectives motivate treating reachability and collision clearance as design constraints rather than post-rationalization.

The tolerances of a modular masonry assembly accumulate through the chain from module manufacture to robot placement. A first-order model separates the independent contributions as

$$\sigma_{tot}^2 = \sigma_r^2 + \sigma_t^2 + \sigma_m^2 + \sigma_p^2$$

where σ_r is robot positioning uncertainty, σ_t is tooling and gripper error, σ_m is module dimensional variability, and σ_p is placement and joint-setting error. The sum of variances is appropriate when the error sources are approximately independent and unbiased. The resulting total tolerance can be compared with the allowable joint width and with the curvature bound from the previous section. If the total tolerance exceeds the joint allowance, the design must either reduce module size, increase joint width, or tighten the fabrication process. This coupling is important because a layout that is geometrically exact but toleranced too tightly is not buildable at scale.

Assembly sequence adds a temporal constraint. Each module must be supported by previously placed modules, temporary supports, or the substrate until the mortar or friction interface can carry load. The sequence must also avoid trapping the robot in enclosed regions or forcing placements that collide with already built geometry. Sequence feasibility can be encoded as a precedence graph in which an edge from module j to module i means that j must be placed before i. The optimizer searches over both the layout and the topological order, rejecting sequences that violate support or access conditions. Because the number of possible sequences grows combinatorially, the sequence is usually optimized jointly with the layout using heuristic or evolutionary search rather than exhaustive enumeration.

5. Generative Optimization Workflow

The design problem is multiobjective. Material volume, fabrication time, surface deviation, and constraint violation compete, and there is no single scalar objective that is universally appropriate. A weighted formulation provides a practical search objective:

$$\min_X J = w_v \frac{V}{V_0} + w_t \frac{T}{T_0} + w_e \frac{E}{E_0} + w_s P_s + w_c P_c$$

where V is material volume, T is estimated fabrication time, E is surface deviation, P_s is a support or sequence penalty, and P_c is a collision or clearance penalty. The terms V_0, T_0, and E_0 normalize each objective by a reference layout, and the weights w_v, w_t, w_e, w_s, and w_c encode design priorities. The penalties are not substitutes for hard constraints; they guide the search when feasible regions are narrow, while final candidates must satisfy the structural and robotic constraints directly.

Robustness is evaluated by introducing a bounded uncertainty vector ξ that collects variations in module dimensions, friction coefficient, and placement error. For each constraint j, let $m_j(X, \xi)$ be a normalized margin that is positive when the constraint is satisfied and negative when it is violated. The robust feasibility margin is

$$R(X) = \min_{\xi \in \Xi} \min_j m_j(X, \xi), R(X) \geq 0$$

where Ξ is the uncertainty set. A design with $R(X) \geq 0$ satisfies all constraints for every realization in Ξ, which is a stronger condition than nominal feasibility. The margin also provides a scalar measure of how close a candidate lies to the feasibility boundary, making it useful for ranking otherwise similar layouts.

The optimization workflow proceeds by generating candidate layouts from a parametric surface decomposition, evaluating geometric fit and curvature bounds, constructing the adjacency and precedence graphs, checking structural equilibrium and robotic reachability, and then applying a multiobjective evolutionary search. Evolutionary algorithms are appropriate because the decision space mixes continuous positions, discrete orientations, and combinatorial sequence variables, and because reachability and collision checks are non-smooth. Non-dominated sorting and crowding-distance selection, as used in NSGA-II, provide a established mechanism for maintaining a Pareto front of layouts with different trade-offs between material, time, and fidelity [13]. The output is not a single optimal structure but a set of feasible candidates whose constraint margins and objective values are explicit.

6. Analytical Assessment and Technical Discussion

The formulation exposes several trade-offs that are difficult to see when geometry, structure, and fabrication are optimized separately. First, curvature-adaptive module sizing reduces geometric deviation but increases the number of modules and joints. More joints increase fabrication time and create more potential slip planes, so the structural feasibility margin may decrease even as surface fidelity improves. The optimizer therefore tends to select larger modules in low-curvature regions and smaller modules near high-curvature transitions, where joint opening and bearing loss are most likely. This behavior is consistent with digital stereotomy and vault-design research, which treats voussoir geometry as a structural and fabrication variable rather than a purely visual subdivision [14].

Second, reachability can dominate structural considerations in deep or enclosed surfaces. A compression-feasible layout may require modules to be inserted from directions that are blocked by the partially built structure. If collision clearance is enforced, the optimizer may prefer open bond patterns, temporary openings, or a different build order. This effect is especially relevant for robotically assembled masonry because the manipulator must approach each module without contacting the existing work. Rapid prototyping and digital materialization research have shown that fabrication constraints often reveal design opportunities rather than merely restricting form [15]. In the present framework, the sequence graph makes those opportunities explicit by treating access as a topological property of the assembly.

Third, tolerance propagation couples material choice to robotic precision. If module variability is large, the joint width must accommodate it, which may reduce the effective bearing area and the friction margin. Alternatively, tighter module tolerances may increase prefabrication cost. The variance model in Equation 8 provides a transparent way to compare these options without pretending that a single tolerance value is universal. The model also clarifies why robust optimization is necessary: a layout that is feasible at nominal dimensions may fail when module dimensions and placement errors are combined. Digital materiality and material synthesis perspectives support this integrated view, in which the material unit, the robotic process, and the geometric system are co-designed [16].

Fourth, the objective weights are not neutral. A design office may prioritize surface fidelity, while a fabrication team may prioritize time and reachability. The weighted sum is useful for generating a Pareto front, but the final choice requires an explicit decision rule. The robust margin R(X) helps because it separates feasibility from preference: a candidate with a large positive margin can absorb more uncertainty, while a candidate near zero is fragile even if its weighted objective is low. This distinction is important in masonry, where small geometric errors can lead to local crushing, rocking, or loss of contact.

The framework also has boundary conditions. The equilibrium model is a lower-bound limit-state formulation and does not predict cracking, crushing, or dynamic response. Mortar curing and time-dependent strength gain are not represented, so sequences that rely on early-age mortar must be checked separately. The reachability model assumes a fixed robot base and known obstacle geometry; mobile manipulators and relocatable bases would require an additional base-placement variable. The curvature bound is approximate and should be refined by contact analysis in detailed design. These limitations do not undermine the generative formulation, but they define the level at which it should be used: as a screening and design-space exploration tool that reduces the number of candidates requiring high-fidelity analysis.

The broader technical significance is that modular masonry can be treated as a coupled discrete design problem rather than a surface to be post-processed. Computational geometry supplies the curvature and adjacency logic, limit analysis supplies the compression-only feasibility test, and robotic constraints supply the kinematic and tolerance logic. When these are combined in a generative search, the resulting layouts are not merely forms that happen to be stable or reachable; they are forms whose stability and reachability have been part of the search. This alignment is particularly relevant for robotically fabricated masonry, where the cost of late-stage redesign is high and where the physical assembly sequence is inseparable from the geometric design.

7. Conclusion

This paper has presented a generative design optimization framework for modular masonry structures that couples computational geometry, compression-only equilibrium, and robotic fabrication constraints. The formulation represents a structure as a variable set of modules with position, orientation, and unit type; enforces equilibrium and friction-cone feasibility through a discrete limit-state condition; constrains module size by surface curvature and joint tolerance; and evaluates inverse-kinematic reachability, collision clearance, payload, and tolerance propagation for robotic placement. A multiobjective search minimizes material volume, fabrication time, surface deviation, and constraint penalties, while a robust margin quantifies feasibility under bounded uncertainty in dimensions, friction, and placement error.

The analytical assessment shows that the main design trade-offs arise from interactions among these constraints rather than from any one constraint in isolation. Curvature-adaptive sizing improves surface fidelity but increases joint count and fabrication time; reachability and collision clearance can override structural preferences in enclosed or deep surfaces; and tolerance propagation links material variability to robotic precision and joint bearing. By making these interactions explicit, the framework provides a basis for computational design tools that treat masonry modules, structural load paths, and robotic assembly as parts of one optimization problem. The result is a more defensible route from geometric intention to

robotically buildable modular masonry than workflows that separate form, structure, and fabrication into sequential stages.

References

1. Pottmann, H., Asperl, A., Hofer, M., & Kilian, A. (2007). Architectural geometry. Bentley Institute Press.
2. Oxman, R. (2006). Theory and design in the first digital age. Design Studies, 27(3), 229–265.
3. Block, P., Ciblac, T., & Ochsendorf, J. (2006). Real-time limit analysis of vaulted masonry buildings. Computers & Structures, 84(29–30), 1841–1852.
4. Kilian, A., & Ochsendorf, J. (2005). Particle-spring systems for structural form finding. Journal of the International Association for Shell and Spatial Structures, 46(2), 77–84.
5. Panozzo, D., Block, P., & Sorkine-Hornung, O. (2013). Designing unreinforced masonry models. ACM Transactions on Graphics, 32(4), Article 91.
6. Deuss, M., Panozzo, D., Whiting, E., & Sorkine-Hornung, O. (2014). Assembling self-supporting structures. ACM Transactions on Graphics, 33(6), Article 214.
7. Krish, S. (2011). A practical generative design method. Computer-Aided Design, 43(1), 88–100.
8. Kolarevic, B. (2003). Architecture in the digital age: Design and manufacturing. Taylor & Francis.
9. Xu, W., Luo, D. A. N., & Gao, Y. (2019, April). Automatic brick masonry system and its application in on-site construction. In 4th International Conference on Computer-Aided Architectural Design Research: Intelligent and Informed,(CAADRIA) (Vol. 1, pp. 83-92).
10. Bock, T. (2007). Construction robotics. Autonomous Robots, 22(3), 201–209.
11. Bock, T., & Linner, T. (2016). Construction robots: Elementary technologies for offsite and onsite automation. Cambridge University Press.
12. Gramazio, F., & Kohler, M. (2008). Digital materiality in architecture. Lars Müller Publishers.
13. Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 6(2), 182–197.
14. Rippmann, M., Lachauer, L., & Block, P. (2012). Interactive vault design. International Journal of Space Structures, 27(4), 219–230.
15. Sass, L., & Oxman, R. (2006). Materializing design: The implications of rapid prototyping in digital design. Design Studies, 27(3), 325–355.
16. Menges, A. (2015). Material synthesis: Fusing the physical and the computational. Architectural Design, 85(5), 8–15.
17. Adriaenssens, S., Block, P., Veenendaal, D., & Williams, C. (Eds.). (2014). Shell structures for architecture: Form finding and optimization. Routledge.
18. Pottmann, H., Eigensatz, M., Vaxman, A., & Wallner, J. (2015). Architectural geometry. Computers & Graphics, 47, 145–164.