Computational Physics Architecture for Automated Panel Bending Validation: The Phase 4 Physics Gatekeeper

1. Introduction: The Deterministic Imperative in Industrial Automation

The transition from Phase 3 (Sequencing) to Phase 4 (Validation & Simulation) in the control pipeline of a Salvagnini P4 Panel Bender represents a fundamental shift in abstraction. While the Phase 3 Sequencer operates in a combinatorial space—optimizing the logical order of bends based on geometric topology and precedence constraints—it largely assumes an idealized physical world. It posits that rotations are instantaneous, manipulators have infinite rigidity, and vacuum grippers provide absolute adhesion. The role of the Phase 4 "Physics Gatekeeper" is to dismantle these assumptions. It subjects the candidate sequence to a rigorous, physically accurate simulation environment that enforces the non-negotiable laws of rigid body dynamics, tribology, and kinematics.

This report details the architectural specification for a C++ computational engine designed to validate the physical feasibility of manufacturing sheet metal components. The objective is to ensure that the instruction set generated for the machine does not result in kinematic singularities, material slippage, or catastrophic collisions. By integrating advanced computational geometry algorithms for grasp planning, tribological models for suction cup interfaces, and heuristic search strategies for repositioning (Repo) logic, we establish a digital twin that guarantees process reliability before physical execution.

1.1 The Operational Context: Salvagnini P4 Kinematics

The Salvagnini P4 is a pinnacle of "Panel Bending 4.0," characterized by its universal tooling and adaptive automation.[1] Unlike traditional press brakes where the operator manages the workpiece, the P4 utilizes a fully automated manipulator that moves the blank on a horizontal plane.[3] The core kinematic chain involves:

- **The Manipulator:** An electric carriage providing $X - Y$ translation and θ rotation, feeding the sheet into the bending line.
- **The Press (Bending Unit):** A C-frame structure housing the upper and lower blades which execute the bend through interpolated motion.[4]
- **The ABA (Automatic Blankholder):** A dynamic tooling system that adjusts its length in masked time to match the bend line geometry.[1]

The validation engine must model these subsystems with high fidelity. The "Job Console" and proprietary "Pack Modus" software manage the high-level production flow [3], but the granular physics validation requires a specialized sub-engine. The P4's ability to handle batch-one production [5] implies that the validation system cannot rely on pre-computed templates; it must simulate novel geometries in real-time.

1.2 System Architecture and Inputs

The Phase 4 engine functions as a filter.

- **Input:** A candidate "Bend Sequence" Directed Acyclic Graph (DAG), where nodes represent machine states (Bend, Rotate, Flip) and edges represent transitions.
- **Constraints:** The machine specification (P4-2120, P4-2520, etc.) defining force limits (F_{max}), torque limits (T_{max}), and kinematic envelopes.[6]
- **Output:** A binary VALID/INVALID flag accompanied by a "Failure Vector" detailing the precise mode of failure (e.g., "Shear Slip at Step 4, $F_{inertial} > F_{friction}$ ").

The architecture is modular, built upon three pillars:

1. **Grasp Validator:** A computational geometry module solving the "Largest Empty Rectangle" problem to find valid suction zones.
2. **Vacuum Physics Engine:** A quasi-static solver calculating shear and normal forces at the pneumatic interface.
3. **Kinematic Planner (Repo):** A path-planning module managing the hand-off between the manipulator and the blankholder.

2. Geometric Modeling of the Machine Environment

The fidelity of the validation is bounded by the accuracy of the virtual machine model. In C++, this is not merely a visual mesh but a parametric volume capable of collision queries.

2.1 The Automatic Blankholder (ABA) Model

The ABA is the most complex static collider in the system. As described in the technical literature, the ABA adjusts its length to clamping requirements.[1] It is composed of modular segments that can be combined in steps (typically 5mm or similar increments).[7] In the C++ architecture, the ABA is modeled as a ParametricTool class.

- **State:** current_length, segment_configuration.
- **Update Logic:** For every bend step i, the engine queries the required bend line length L_{req} .

$$L_{ABA} = \text{floor}(L_{req}/\text{step\_size}) \times \text{step\_size}$$

The gap $G = L_{req} - L_{ABA}$ represents the unsupported flange width.

- **Validation Check:** If $G > G_{max}$ (material specific limit), the engine flags a "Quality Risk" (droop). If $L_{ABA} > L_{req}$, it flags a "Collision" (tool hitting side flanges).

2.2 The Manipulator and Suction Unit

The manipulator's transport mechanism relies on a suction cup array. The arrangement is machine-specific (e.g., P4-2520 has a larger reach than P4-2120 [6]). The C++ model must represent the "dead zones" of the manipulator—areas where the gripper cannot reach or where the suction cups would overhang the table. The suction unit includes a "Rotator".[3] The geometric model must account for the rotator's center. Unlike a simple Cartesian gantry, the P4 rotation can be eccentric or concentric depending on the specific model configuration. The validation engine must compute the swept volume of the sheet during rotation to ensure it does not strike the C-frame columns.[8]

2.3 The Workpiece: Mesh Topology and Evolution

The sheet metal part is a dynamic entity. A static STL file is insufficient because the topology changes (faces merge, edges split) with every bend. We utilize a **Half-Edge Data Structure (DCEL)** to represent the mesh.

- **Topological Queries:** The DCEL allows constant-time $O(1)$ access to adjacent faces, which is critical for checking "Self-Collision" during folding.
- **Mass Property Integration:** As the part bends, its Center of Gravity (CoG) and Inertia Tensor I shift. We implement a dynamic update using the Divergence Theorem, integrating over the mesh surface to update mass properties after every deformation step. This is vital for the Vacuum Physics module, as the lever arm for torque calculations changes constantly.

$$I_{new} = I_{old} + \Delta I_{bend}$$

3. Algorithmic Grasp Validation

The primary interaction between the machine and the material is the grasp. The "Grasp Validation" module answers a critical question: *Is there a valid surface area on the current top face of the sheet that can accommodate the gripper without leakage?*

3.1 The Largest Empty Rectangle (LER) Problem

We model the sheet face as a general polygon P with n vertices, potentially containing holes (cutouts) and reflex vertices (notches). The gripper footprint is a rectangle R_{grip}. The problem reduces to finding the **Maximum Area Axis-Aligned Rectangle** inscribed in P.

3.1.1 Theoretical Basis

The problem of finding the largest empty rectangle in a polygon with holes is a classic problem in computational geometry.[9] A naive grid-based approach (rasterizing the polygon) [12] is computationally expensive for high-resolution requirements and suffers from aliasing. Instead, we employ an exact vector-based algorithm. The standard algorithm operates in $O(n \log^2 n)$ or $O(n \log n)$ time using a divide-and-conquer strategy or a sweep-line approach. The core insight is that the maximal rectangle must be bounded by "obstacles" (polygon edges or hole vertices) on all four sides.[9]

3.1.2 The Sweep-Line Algorithm Implementation

Our C++ implementation utilizes a specialized version of the plane sweep algorithm adapted from the principles in [11] and.[13]

1. **Event Generation:** We decompose the polygon boundaries and hole edges into a set of vertical segments. The x-coordinates of these endpoints define the "events" for the sweep line.
2. **Active Set Maintenance:** As the sweep line moves from x_{min} to x_{max}, we maintain a 1D data structure (a Segment Tree) representing the "free" intervals along the current scan line.
3. **Interval Processing:** At each event x_i, the Segment Tree is updated. If an edge starts, an interval is blocked. If an edge ends, an interval opens.
4. **Candidate Generation:** For every continuous free interval y_{start} to y_{end} on the sweep line, we compute the potential rectangle extending back to the point where this interval was first created.
5. **Maximization:** We compare the area of each candidate against a running max_rect.

We leverage the **CGAL (Computational Geometry Algorithms Library)**, specifically the Largest_empty_iso_rectangle_2 class [14], to handle the robust geometric predicates. Since CGAL's default implementation focuses on point sets, we adapt it by discretizing the polygon edges into "constraint points" or by using the General_polygon_set_2 traits to handle boolean operations (Sheet minus Holes).

3.2 Discrete Suction Cup Mapping

Finding the LER is the first step. The second is mapping the physical suction cups to this rectangle. The P4 gripper is not a single continuous pad but an array of discrete cups (e.g., a 4×2 matrix).

- **The "Leakage" Heuristic:** Even if the gripper frame fits in the LER, individual cups might land on small holes or be too close to an edge.
- **Ray-Casting Check:** For the candidate grasp pose T_{grasp}, we perform a ray-cast from the center of each suction cup C_i onto the mesh.
  - If RayCast(C_i) hits a face $\rightarrow$ **Cup Active**.
  - If RayCast(C_i) hits nothing (hole) $\rightarrow$ **Cup Inactive** (Vacuum Loss).
  - If Distance(C_i, Edge) < Margin $\rightarrow$ **Cup Inactive** (Edge Leakage).

The grasp quality Q is defined as the sum of active cup areas:

$$Q_{grasp} = \sum_{i=1}^{N} \delta_i \cdot A_{cup}$$

where $\delta_i \in \{0,1\}$. The sequence is valid only if $Q_{grasp} > A_{threshold}$, where $A_{threshold}$ is derived from the physics requirements (Section 4).

3.3 Dynamic Surface Evolution and Dead Zones

The graspable area is not static. After a bend, a portion of the sheet becomes a flange (vertical).

- **Feature Erosion:** The validation logic must "erode" the available polygon P by the manipulator's safety clearance. Up-bends create "walls" that the manipulator cannot touch.
- **Machine Limits:** The P4 has specific "operating limits" [8] where the manipulator cannot travel. These are treated as "Virtual Holes" in the LER algorithm. For example, the snippet [8] mentions "Areas that cannot be punched because outside manipulators' X travel range." These same limits apply to grasping. The LER search space is the intersection of the Sheet Face and the Machine Kinematic Range.

4. Vacuum Physics Simulation: The Tribological

Interface

Geometric fit does not guarantee physical hold. The most common failure in automated bending is the part slipping during high-speed transport. The Phase 4 engine must simulate the forces at the suction interface.

4.1 The Tribological Model: Friction and Shear

The holding force of a vacuum cup is anisotropic. It is strong in the normal direction (suction) but weaker in the tangential direction (friction). The maximum shear force F_{shear}^{max} before slip occurs is governed by the Coulomb friction model [16]:

$$F_{shear}^{max} = \mu \cdot F_{normal}$$

$$F_{normal} = (P_{atm} - P_{vac}) \cdot A_{active} \pm mg$$

4.1.1 The Friction Coefficient (μ)

Crucially, industrial sheet metal is rarely clean. It is often coated with mill oil or protective films.

- **Dry Rubber vs. Steel:** $\mu \approx 0.6 - 0.8$.
- **Oiled Steel:** $\mu \approx 0.1 - 0.2$.[16] The simulation defaults to the conservative "Oiled" value. If the validation assumes a dry coefficient, the simulation will pass unsafe sequences that result in machine crashes.

4.2 Dynamic Load Simulation

The manipulator does not move at constant velocity; it accelerates. The Salvagnini P4 is a high-dynamic machine (accelerations $> 1G$).

The inertial force $F_{inertial}$ acting on the sheet is:

$$F_{inertial} = -m \cdot \mathbf{a}_{manipulator}$$The validation check is:$$||F_{inertial}|| < \sum_{i \in Active} \mu_i \cdot F_{normal, i}$$

4.3 Moment Resistance and Rotational Slip

A single suction cup acts as a pivot. It offers zero resistance to rotation (Yaw) in the plane of the sheet. Resistance to rotation comes from the *distribution* of the cups.[17] When the manipulator rotates the sheet (angular acceleration α), or when linear acceleration acts on a

CoG offset from the grasp center, a Torque τ is generated.

\tau_{net} = I_{zz} \alpha + (r_{CoG} \times F_{linear})$$The resistive torque of the gripper τ_{resist} is the sum of the shear resistance of each cup multiplied by its lever arm from the Instantaneous Center of Rotation (ICR).$$\tau_{resist} = \sum_{i} (\mu F_{N,i}) \cdot ||r_i - r_{ICR}||

If $\tau_{net} >$, the sheet will spin in the gripper, losing registration. This is a critical failure mode. The C++ engine calculates the "Grasp Stability Margin" ($GSM = \tau_{resist} / \tau_{net}$). If $GSM < 1.2$ (20% safety factor), the step is flagged as invalid.

4.4 Soft-Body vs. Rigid-Body Approximations

Full soft-body simulation (FEM) of the rubber cup [18] is too slow for real-time sequence validation. We employ a **Lumped Parameter Model**.

- **Spring-Damper Logic:** Each cup is modeled as a stiff spring in the normal direction and a slip-stick element in the tangential direction.
- **The "Peel" Effect:** We monitor the distribution of normal forces. If the moment on the cup array causes the normal force on the "leading edge" cups to drop below zero (tension), it indicates a "Peel" condition where the cup is lifting off. Vacuum is lost immediately in that cup, shifting the load to remaining cups, potentially causing a cascade failure. This phenomenon is modeled by iteratively solving for the equilibrium of the rigid sheet under the constraints of the unilateral spring contacts (cups can push, but only pull up to F_{vac}).

5. Repositioning (Repo) Logic: The Strategic Solver

"Repo" (Repositioning) allows the machine to process parts larger than its reach or to change grip for geometric clearance. It is a complex handover maneuver involving the Manipulator, the Sheet, and the ABA.[19]

5.1 The Strategic Necessity of Repo

A Repo is triggered when:

1. **Kinematic Limits:** The target position for the next bend X_{bend} places the manipulator outside its travel limits ($X_{man} < X_{min}$ or $X_{man} > X_{max}$).[8]
2. **Grasp Interference:** The current grasp position covers the bend line of the next operation.
3. **Torque Overload:** The CoG has shifted so far that the current grasp is unstable (as defined in Section 4.3).

5.2 The Repo Solver Algorithm

The Repo Solver must find a *transition* plan. This is an optimization problem: Find a new grasp pose P_{new} that is valid and maximizes future utility. We implement an *A Search Algorithm** [19] to solve this.

5.2.1 State Space and Cost Function

- **State:** $(x, y, \theta)_{grip}$.
- **Heuristic ($h(n)$):** We use a "Look-Ahead" heuristic. We project the sequence forward K steps. The heuristic cost is inversely proportional to the number of future steps that can be executed *without* another repo.

$$h(n) = \frac{1}{\text{StepsWithoutRepo}(P_{new})}$$

- **Cost ($g(n)$):** The time cost of the move.

$$g(n) = \frac{||P_{old} - P_{new}||}{v_{max}} + T_{clamp\_cycle}$$

5.2.2 The Handover Constraint (Pigeonhole Principle)

During Repo, the Manipulator releases the sheet. The sheet must be held by the ABA (Blankholder).

- **Constraint:** There must be a "flat" zone on the sheet under the ABA that is free of flanges and holes.
- **Algorithm:** We run a 1D version of the LER algorithm on the strip of the sheet currently under the press. We identify valid "Clamping Zones." The Repo is only possible if:

$$\text{ValidClampZone} \cap \text{MachineCenter} \neq \emptyset$$

  If the sheet cannot be clamped safely (e.g., due to a cutout under the pads), the Repo is impossible, and the sequence is invalid.

5.3 Dynamic Collision During Rotation

Repositioning often involves rotating the part. The most dangerous phase is the rotation, where the corners of the part swing through a large arc.

- **Swept Volume Check:** We compute the convex hull of the part. We then extrude this hull along the rotational arc to create a "Swept Volume" mesh.

- **FCL Integration:** We use the **Flexible Collision Library (FCL)** [21] to check for intersections between this Swept Volume and the machine's static geometry (C-frame columns).
- **Retraction Logic:** If a collision is detected, the solver attempts to find a "Retracted Rotation" strategy:
  1. Move X to safe zone.
  2. Rotate θ.
  3. Move back.
     This adds to the cycle time but saves the part.

6. Collision Detection Systems

The collision engine is the final fail-safe. It prevents the "Crash."

6.1 Hierarchy Selection: Dynamic AABB Trees

For the sheet metal part, we use **Dynamic AABB (Axis-Aligned Bounding Box) Trees**.[22]

- **Why AABB?** Sheet metal parts are predominantly orthogonal (box shapes). AABBs fit these shapes tightly.
- **Update Strategy:** When a bend occurs, we do not rebuild the tree. We perform a "Refit" operation. The branch of the tree corresponding to the bent flange is rotated, and the parent boxes are resized. This allows for $O(1)$ collision updates during the bending animation.

6.2 The "Height Check" (2.5D Collision)

A specific failure mode in panel bending is vertical interference. As flanges are bent up, they may hit the upper tool holder or the manipulator carriage.[24] Full 3D collision is expensive. We implement a fast **2.5D Height Map Check**.

1. **Projection:** We project the sheet geometry onto the XY plane.
2. **Z-Buffering:** We assign a Z_{max} value to each polygon face.
3. **Ceiling Check:** We compare this Z_{max} against the "Ceiling Map" of the machine (the underside of the press beam).

$$\text{Collision} \iff \exists(x, y) : Z_{sheet}(x, y) > Z_{machine}(x, y)$$

This rapidly filters out 90% of potential collisions (e.g., box-closing bends) without expensive mesh-mesh intersection tests.

6.3 Self-Collision and Geodesic Filters

Complex bends (hems, returns) can cause the sheet to fold into itself.

- **The Problem:** Adjacent faces in a mesh always "touch" at their shared edge. A naive collision check would flag this as a crash.
- **The Solution:** We use a **Geodesic Distance Filter**. We only check collisions between face pairs (F_i, F_j) where the geodesic distance (distance along the surface) $D_g(F_i, F_j) > \epsilon$. This effectively ignores the "hinge" while catching the "penetration" of distal flanges.

7. Implementation Strategy and Simulation Loop

The Phase 4 engine is implemented as a high-performance C++ service.

7.1 Class Structure and Libraries

- **Core Libraries:**
  - **CGAL:** For LER grasp validation and boolean operations on polygons.[13]
  - **FCL (Flexible Collision Library):** For broad-phase and narrow-phase collision detection between the dynamic sheet and static machine.[21]
  - **Eigen:** For high-speed linear algebra (Inertia tensors, coordinate transformations).
- **Architecture:**

```C++
class PhysicsGatekeeper {
    GraspValidator grasp_validator; // CGAL-based
    VacuumEngine vacuum_physics;    // Custom Tribology Solver
    RepoSolver repo_planner;        // A* Path Planner
    CollisionSystem collision_sys;  // FCL Wrapper

  public:
    ValidationResult Validate(BendSequence& seq);
};
```

7.2 The Simulation Loop (Discrete Event Simulation)

The validation does not run in real-time but in "logical time."

1. **Load Step i:** Apply topological changes (Folding). Update Mass Props.
2. **Transport Phase:**
   - Calculate Path $P_{start} \to P_{bend}$.

- **Vacuum Check:** Solve dynamic equilibrium ($F_{inertial}$ vs μF_{vac}).
- **Collision Check:** FCL query on Swept Path.

3. **Positioning Phase:**
   - **Kinematic Check:** Is P_{bend} within (X_{min}, X_{max}) ?
   - If NO $\rightarrow$ Trigger RepoSolver.
4. **Bending Phase:**
   - Animate bend (0 to α degrees).
   - **Height Check:** 2.5D Z-buffer check.
   - **Self Check:** Geodesic-filtered mesh intersection.
5. **Advance:** $i \leftarrow i + 1$.

7.3 Parallelization

To achieve rapid validation (sub-second for typical parts), we utilize **OpenMP** to parallelize the "Ray-Cast" operations in the Grasp Validator and the "Leaf-Node" checks in the Collision System. The LER algorithm, being sweep-line based, is inherently sequential but can be run in parallel for different candidate grasp orientations.

8. Conclusion

The "Physics Gatekeeper" is the bridge between the digital abstraction of a bend sequence and the unforgiving reality of steel manufacturing. By implementing rigorous algorithms—specifically the **Largest Empty Rectangle** for grasp planning, **Tribological Shear Models** for vacuum validation, and *A Heuristic Search** for repositioning—we ensure that the Salvagnini P4 operates within its physical limits.

This architecture moves beyond simple geometric interference checking. It simulates the invisible forces—friction, inertia, torque—that are the primary causes of process failure in automated fabrication. The resulting C++ engine provides the "Feasibility Guarantee" required for true Lights-Out Manufacturing, ensuring that every program sent to the machine is not just geometrically possible, but physically robust.

Structured Data Appendix

Table 1: Algorithm Selection Matrix for Validation Subsystems

Subsystem	Problem Domain	Selected Algorithm	Computational Complexity	Justification

Grasp Region	Comp. Geometry	Sweep-Line with Segment Tree	$O(N \log N$	Handles non-convex polygons with holes efficiently; superior to grid rasterization.[11]
Repo Planning	Path Planning	A* Search with Look-Ahead	$O(b^d$ (Heuristic)	Finds optimal grasp transition minimizing cycle time; prevents local minima.[20]
Collision	Intersection	Dynamic AABB Tree (FCL)	$O(\log N$ query	Optimized for orthogonal sheet metal shapes; fast "Refit" operations.[23]
Vacuum	Statics/Dynamics	Quasi-Static Equilibrium	$O(1$	Sufficient for process validation; faster than full FEM while capturing slip failure.[25]

Table 2: Validation Failure Modes and Physics Checks

Failure Mode	**Physics Domain**	**Mathematical Check**	**Mitigation Strategy**
Shear Slip	Tribology	$F_{inertial} > \mu \cdot$	Reduce acceleration (a_{max}); Add suction cups.

Rotational Slip	Mechanics (Moments)	$\tau_{net} >$	Reposition grasp closer to CoG; Increase cup spread.
Box Collision	Geometry (2.5D)	Mesh $Z_{max} >$ Machine Throat	Change bend sequence (reorder flanges).
Vacuum Leak	Pneumatics	Grasp area $\cap$ Hole $\neq$	Shift grasp point via LER algorithm.
Repo Deadlock	Logic/Planning	$\text{ValidClamp} =$	No solution; part requires manual intervention or design change.

Works cited

1. Panel Bender -P4 - MachineryHost, accessed February 3, 2026, https://f.machineryhost.com/71b2fab0316bfa7fa24ce0a0e5460bad/Brochure%20-%20Panel%20Bender%20-P4.pdf
2. Panel benders, Panel bender machines - Salvagnini, accessed February 3, 2026, https://www.salvagninigroup.com/en-INT/products/panel-benders
3. p4-benders.pdf - Empire Machinery, accessed February 3, 2026, https://www.empire-machinery.com/wp-content/uploads/2021/06/p4-benders.pdf
4. Panel Bending Tools - JEELIX, accessed February 3, 2026, https://www.jeelix.com/panel-bending-tools/
5. Salvagnini panel bending: P4-2520 bends 4 different parts - YouTube, accessed February 3, 2026, https://www.youtube.com/watch?v=gOMHcSt2yQ0
6. Automatic Sheet Metal Bending Machine - Salvagnini P4, accessed February 3, 2026, https://www.salvagninigroup.com/en-US/products/panel-benders/P4
7. The automatic hybrid adaptive Panel Bender. - Sanson Machinery Group, accessed February 3, 2026, http://www.sansonmachinery.com/wp-content/uploads/2013/12/P4Xe-UK-3962020312.pdf
8. PUNCHING SYSTEMS, TOOLS AND ACCESSORIES, accessed February 3, 2026, https://irp.cdn-website.com/c8f65e13/files/uploaded/Salvagnini+Punching+Tools+Catalog.pdf
9. Get Largest Inscribed Rectangle of a Concave Polygon - MathOverflow,

accessed February 3, 2026, https://mathoverflow.net/questions/105837/get-largest-inscribed-rectangle-of-a-concave-polygon
10. Finding the largest rectangle in several classes of polygons - DASH (Harvard), accessed February 3, 2026, https://dash.harvard.edu/server/api/core/bitstreams/7312037d-c0a3-6bd4-e053-0100007fdf3b/content
11. Maximum-Area Rectangles in a Simple Polygon, accessed February 3, 2026, https://d-nb.info/1366001346/34
12. Algorithm to find the largest empty rectangle amid other polygons - Stack Overflow, accessed February 3, 2026, https://stackoverflow.com/questions/27400329/algorithm-to-find-the-largest-empty-rectangle-amid-other-polygons
13. Largest Empty Rectangle - CGL at Tel Aviv University Portal - Computational Geometry Lab, accessed February 3, 2026, https://www.cgl.cs.tau.ac.il/projects/largest-empty-rectangle/
14. CGAL 6.1.1 - Inscribed Areas: User Manual, accessed February 3, 2026, https://doc.cgal.org/latest/Inscribed_areas/index.html
15. CGAL::Largest_empty_iso_rectangle_2
16. (PDF) Physics of suction cups - ResearchGate, accessed February 3, 2026, https://www.researchgate.net/publication/336606547_Physics_of_suction_cups
17. Learning Robotic Grasping Policies for Suction Cups and Parallel Jaws in Simulation - EECS at Berkeley, accessed February 3, 2026, https://www2.eecs.berkeley.edu/Pubs/TechRpts/2019/EECS-2019-119.pdf
18. Constraint-based Simulation of Passive Suction Cups | Professor Sheldon Andrews, accessed February 3, 2026, http://profs.etsmtl.ca/sandrews/publication/suction_tog2022/
19. Automated Process Planning for Sheet Metal Bending Operations - Carnegie Mellon University's Robotics Institute, accessed February 3, 2026, https://www.ri.cmu.edu/pub_files/pub1/kim_kyoung_k_1998_1/kim_kyoung_k_1998_1.pdf
20. A Method on Dynamic Path Planning for Robotic Manipulator Autonomous Obstacle Avoidance Based on an Improved RRT Algorithm - MDPI, accessed February 3, 2026, https://www.mdpi.com/1424-8220/18/2/571
21. FCL: A General Purpose Library for Collision and Proximity Queries - GAMMA, accessed February 3, 2026, https://gamma.cs.unc.edu/FCL/fcl_docs/webpage/pdfs/fcl_icra2012.pdf
22. Collision detection during planning for sheet metal bending by bounding volume hierarchy approaches | Request PDF - ResearchGate, accessed February 3, 2026, https://www.researchgate.net/publication/324742687_Collision_detection_during_planning_for_sheet_metal_bending_by_bounding_volume_hierarchy_approaches
23. On sheet metal unfolding. Part 12: searching for feasible bends | by Analysis Situs | Medium, accessed February 3, 2026, https://analysis-situs.medium.com/on-sheet-metal-unfolding-part-12-searching-for-feasible-bends-2a91a8976431

24. Collision Detection Algorithm for 3D Simulation of Sheet Metal Bending, accessed February 3, 2026, https://qikan.cmes.org/zhjc/EN/10.13462/j.cnki.mmtamt.2021.11.020
25. Simulation Model for Robotic Pick-Point Evaluation for 2-F Robotic Gripper - MDPI, accessed February 3, 2026, https://www.mdpi.com/2076-3417/13/4/2599