# Engineering & CAD Skills Library

Verbatim skill rules for AI agents: CAD/Onshape, 3D printing, mechanical design, FRC, simulation, design process. Personal details redacted.

# Contents


### Design process

1. design-concept-gate
2. engineering-cad-release-workflow
3. engineering-math-proofs
4. custom-part-design-from-requirements
5. cad-part-modeling-practice-drills

### Onshape CAD

1. onshape-api-rate-limits
2. onshape-assemblies-and-mates
3. onshape-gui-fast-recipes
4. onshape-hole-feature-and-standard-content
5. onshape-in-context-and-master-model
6. onshape-large-assembly-performance
7. onshape-native-api-builds
8. onshape-part-modeling-workflow
9. onshape-professional-drawings
10. onshape-quickstart-lessons-log
11. onshape-release-and-versioning
12. onshape-sheet-metal
13. onshape-sketching-and-constraints
14. onshape-surfacing
15. onshape-top-down-master-layout
16. onshape-variables-and-configurations

### 3D printing

1. 3d-print-fits-and-tolerances
2. 3d-print-strength-optimization

### Mechanical design

1. mech-design-bolted-joints-and-fasteners
2. mech-design-fits-and-tolerances
3. mech-design-shafts-bearings-gears-belts

### FRC

1. frc-competition-readiness
2. frc-defense-and-robustness
3. frc-dfm-manufacturing
4. frc-mechanism-design-review
5. frc-prototyping-and-iteration
6. frc-reliability-testing
7. frc-robot-cad-organization
8. frc-robot-design-playbook
9. frc-subsystem-arms-wrists-and-pivots
10. frc-subsystem-climbers-and-endgame

11. frc-subsystem-drivetrain
12. frc-subsystem-elevators-and-telescopes
13. frc-subsystem-intakes
14. frc-subsystem-shooters-and-flywheels
15. frc-systems-engineering
16. frc-weight-and-cg-management
17. industrial-reliability-design-for-frc

### Simulation

1. load-path-simulation-for-assemblies
2. onshape-simulation-fea-sanity

### Agent tooling

1. cloud-agent-prompt-builder

# Design process

# design-concept-gate

> **Summary (description):** MANDATORY before committing to any design concept (structure, bracket, printed part, mechanism, competition challenge, FRC subsystem): list only the real constraints, challenge every self-imposed one, rough out and hand-calc 3-6 fundamentally different forms, and pick on numbers. Run it before any CAD, sizing or refinement.

<!-- changelog (bridge retro, 2026-10-28): 1. §1: confirm the test fixture (gap vs total length, supports, load attachment) and size over the range. 2. §6: model the competitor with the same calculator and beat its governing failure by >=15 %. 3. New §10: hard-limit margin rule + pointer to engineering-cad-release-workflow. -->

# Design concept gate (run BEFORE any CAD or refinement)

Why this exists: a printed-bridge challenge (fixed length 228.6 mm, max 30 g, centre point load) was lost because the designer quietly assumed a low flat truss (35 mm tall), read "design the joints" as "must be many parts", fixed a buckling problem with extra parts instead of deeper members, and optimised panel counts inside one truss type without ever comparing forms. A competitor's single-piece king-post truss, 65 mm tall with 14 mm deep members printed flat, was about 2-3x stronger at a similar mass. Every one of those mistakes is caught by the steps below. Do not skip any step for important CAD. Write the outputs down in the design doc; the gate is passed only when the table in step 5 exists.

## 1. Real constraints only

Copy the rules word for word. Make two lists: - **Hard limits (stated):** e.g. length = 228.6 mm, mass <= 30 g, material, no glue or fasteners, one Part Studio. - **Free variables (NOT stated):** height, width, depth, number of parts, shape, print orientation, where the load attaches, how supports are contacted. Anything not explicitly limited is a free variable and a design lever. If a limit is unclear (e.g. max height, support width, load fixture), ask, or design for the worst plausible case and say so. Never invent a limit silently.

**Test fixture (confirm before sizing):** span or gap vs total part length, support shape/width, how the load attaches and is applied, how the limit is measured (slicer estimate vs scale, brim counted?). If unconfirmed, write a range (e.g. gap 7–8 in) and size for every value in it.

## 2. Challenge every assumption

For each of these, write one line saying whether the rules force it or you assumed it: - "It should look like a normal X" (flat bridge, rectangular bracket). - Envelope: height, width, depth. Ask "what if this doubled?" - Part count. A deliverable called "joints" or "assembly" does not mean you need many parts. Fewer parts and fewer joints is usually stronger and lighter; use joints only where the rules or the print bed force a split. - Load case: point vs distributed, hanging vs bearing, where it attaches. Design for the actual test, not a generic one. - Print or manufacturing orientation is a STRUCTURAL decision (layer lines, deep sections printed flat, walls along the load), not a slicing afterthought.

## 3. Generate 3-6 fundamentally different forms

Not variants of one idea. For a span or bridge, at minimum: tall triangle or king-post / queen-post truss, tied arch (bowstring), deep truss (Pratt/Warren/Howe) at the MAX useful height, deep I-beam or box beam,

suspension or cable-stayed if tension members are allowed. For brackets and parts: solid plate vs ribbed vs truss-pocketed vs shell vs gusseted. Sketch each on paper or as a quick 2D layout.

## 4. Ten-minute hand calc for each form

Same load, same mass budget, same material. For each form estimate: - Member forces from statics. For a point load P at mid-span with height h and half-span a: chord or rafter force about P*a/(2h). Height is the strongest lever; forces drop in proportion to it. - Each compression member: Euler buckling in-plane AND out-of-plane, P_cr = pi^2*E*I/(K*L)^2, using the unbraced length in each direction. - Each tension member: area x allowable stress, with a stress-concentration factor (Kt 1.5-2) at joints, notches, holes and re-entrant corners. - Bearing at supports and at the load point. - Mass from the sliced volume (walls and infill), not just solid CAD. - Label every input FACT, INFERENCE or GUESS. Use `engineering-math-proofs` for the formulas and `3d-print-strength-optimization` for printed-material knock-downs.

## 5. Decision table (required output)

| Form | Governing failure | Predicted capacity | Mass | Parts/joints | Print risk | Score | Pick the winner on predicted capacity per gram, then robustness (how bad is the weakest single feature, e.g. one thin wall). If the winner breaks an assumption you held at the start, that is a good sign, not a problem.

## 6. Red-team the winner before CAD

Ask: "What would a smarter competitor build?" and "What does the strongest known solution to this class of problem look like?" (search past competition winners, e.g. balsa or printed bridge contests). If a competitor's likely design beats yours on the numbers, switch now.

**Competitor model (required):** put the best known/competitor design through the SAME calculator as a row in the step 5 table, name its governing failure, and make sure your winner beats it by >=15 % there. Do this before CAD, not after losing.

## 7. Only then refine

After the gate: size members, then joints, then CAD (Onshape, built-in features only), then the print strategy. Re-run the step 4 calc on the final geometry.

## Quick checklist (paste into the design doc)

- [ ] Rules copied; hard limits vs free variables listed
- [ ] Every self-imposed assumption written and challenged (height, part count, look, load case, orientation)
- [ ] At least 3 fundamentally different forms roughed out
- [ ] Hand calc for each: statics, buckling both planes, tension with Kt, bearing, sliced mass
- [ ] Decision table filled; winner chosen on numbers
- [ ] Red-team: compared against the best-known or competitor solution

## 8. Post-design review checklist (from an independent critique of the lost bridge)

Run every item on the final design. Each one was a real failure. **Engineering** 1. Shape follows the force pattern. A midspan point load gives a triangular moment, so use a king-post, arch or tapered depth, not constant depth carrying dead mass at the ends. 2. Use the full available depth. Chord force is about M/h, so

half the depth means double the force. 3. Size every member to its own force so all of them fail at about the same load. No rule-of-thumb thicknesses (e.g. "tension 2 mm, compression 4 mm"). Check the tension chord especially; it is often the smallest and most critical. 4. Prefer one deep plane over several thin ones. Buckling scales with thickness cubed, so one 14 mm truss beats two 4 mm trusses of the same mass. 5. If you do use multiple planes, brace them properly: top AND bottom, with X-bracing, and rigid joints, not loose press-fits. 6. Apply the load low. A load hung below the bottom chord centres itself; a load on top above narrow feet amplifies any lean. 7. Keep bending members out of the load path (no loading beam with a hole in the middle). Send the load straight into axial members. 8. Fillet every joint and re-entrant corner. Printed PETG is notch-sensitive, and tension joints crack first. 9. Minimise parts and press-fit joints. Every joint adds slop and a failure point. 10. Make the feet and bearings wide enough for bearing area and sideways stability. 11. Check every member that carries a large share of the load (e.g. a centre post) and size it for that. **CAD practice** 12. Use patterns, mirrors and variables for repeated features, never N copies. 13. Delete leftover clutter (empty Part Studio 1, empty Assembly 1, empty parts list). 14. Build an assembly and check fits and interference for every joint. Visually confirm parts seat. 15. Create a named release version (not just "Start"). 16. Drive the key decisions with variables (member sizes tied to forces, span, load), not just height. **Process** 17. Put the trade study in writing (step 5 table): arch vs truss, one plane vs two, load hung vs on top. 18. Check every member's force against its capacity, and name the weakest link and the margin on each member. 19. Check sideways and lateral-torsional stability explicitly. 20. Reinforce the weak link on purpose, then re-check the new weakest link.

## 9. Double- and triple-check (mandatory for important CAD)

- **Check 1, the numbers:** re-run every hand calc from scratch on the FINAL geometry (not the plan), with a second method where possible (e.g. method of joints vs method of sections, or a quick frame solver).
- **Check 2, the CAD through the GUI:** open the document in the Onshape GUI and look at it, don't just trust API output. Rotate the iso, front, top and section views; use Measure on every critical thickness, depth and clearance; run Mass properties on all parts; check the assembly mates and interference; scan the feature list for errors or red features; confirm no clutter and that a release version exists. Screenshot the key views.
- **Check 3, adversarial review:** go through the section 8 checklist line by line as if critiquing a competitor's design, and write the top 3 ways it fails. Fix anything that fails before delivering.
- Report each check's result in the delivery. Never deliver important CAD with an unchecked item.

## 10. Hard limits and release

Design to <= 90 % of any hard limit using a pessimistic estimate; a real measurement (e.g. real slice in the target slicer) is a release gate. Full phased gates: `engineering-cad-release-workflow`.

# engineering-cad-release-workflow

**Summary (description):** MASTER workflow for any design challenge, competition part, printed part or CAD deliverable built from requirements (bridges, brackets, FRC parts, "design X that holds Y under Z g"): use from the moment the rules arrive until the file is delivered. Gated phases: requirements and test fixture, concept trade study plus competitor model, sizing with margins, hard-limit budgets with a real slice, Onshape native build, automated geometry checks, GUI triple check, metadata, test plan, release and delivery.

# Engineering & CAD release workflow (requirements -> delivered file)

Born from a printed-bridge challenge that was lost first time (wrong form, invented limits) and nearly failed second time (sliced mass 31.09 g vs a 30 g limit after predicting 27.1 g; foot nub only seen by eye; material lost on regeneration). Every gate below blocks one of those failures. **Gates are sequential. If a gate fails: stop, fix, re-run that gate and every later gate already done.** Write gate results into the design doc (`DESIGN.md`) as you go; label every number FACT / INFERENCE / GUESS.

## Gate 0 — Requirements, real limits, test fixture

Do: copy the rules verbatim; split hard limits vs free variables (see `design-concept-gate` §1). Pin down the **test fixture**: span/gap vs total length, support shape and width, how the load attaches (hook, bucket, plate), loading rate, how mass is measured (sliced estimate? scale weight? brim counted?), slicer + settings, who prints. Pass: every hard limit has a number and a measurement method; the gap/fixture is confirmed by the user or a range is written ("gap 7–8 in") and all later calcs cover the whole range. Stop: any limit you "assume" that the rules did not state -> ask, or design for the worst case and say so.

## Gate 1 — Concept gate + competitor model

Run `design-concept-gate` in full (3–6 forms, hand calcs, decision table, red-team). Add: model the best known / competitor design **with the same calculator** and name its governing failure; your winner must beat it by a margin on that same model (aim >=15 %). Pass: decision table exists; winner chosen on capacity per gram; competitor row present. Stop: no table -> no CAD.

## Gate 2 — Analytic sizing with margins

Size every member to its own force (statics, buckling both planes, tension x Kt 1.5–2 at nodes, bearing). Use a script (e.g. `analysis/model.py`) so it can be re-run on final geometry and every fixture in the Gate 0 range. Report weakest link + margin for each member. Pass: all members within ~20 % of the same failure load; capacity table over the whole fixture range.

## Gate 3 — Hard-limit budgets (mass, size, time) with calibration discipline

- Design target <= **90 %** of any hard limit (30 g limit -> design to <= 27 g **pessimistic**).
- Use the **pessimistic** estimate, never a factor that makes the number smaller unless it was calibrated on the *same geometry class* (similar member thickness vs wall thickness) and same slicer settings. A

calibration from 2 different parts is not a calibration.
- Printed thin members: 3 walls @0.42 mm ≈ 2.5 mm per side, so members narrower than ~5 mm slice essentially **solid**. Estimate per member: solid where thin, shells + infill % where thick, + top/bottom layers. Brim/skirt/supports: decide explicitly whether they count.
- Report the estimate as a range, upper end against the limit. Pass: upper end of the range <= 90 % of limit. Release gate (Gate 8): a REAL slice in the target slicer (e.g. Bambu Studio, run by the user or via GUI) with the exact rule settings, reading model-only mass. **Never deliver "ready to print" with an unverified hard-limit metric**; say "estimated X–Y g, must be confirmed in the slicer before printing" instead.

## Gate 4 — Onshape build (native only)

One top-down Part Studio, variables drive key sizes, native features only (**never FeatureScript**). API/console builds: follow `onshape-native-api-builds` and `onshape-api-rate-limits` (>=3 s between calls, <=20/min). Build on a plane you will view easily (profile on Front if you want view-cube Front to show the profile). Use clean outlines: union members as solids/regions, then fillet edges with a Fillet feature — avoid polyline offsets that overlap at nodes. Pass: no red features; part count as planned; no clutter tabs.

## Gate 5 — Automated geometry checks (before any human look)

- Profile (pre-build or from the plan): `python3 /home/box/agent-data/workflows/engineering-cad-release-workflow/tools/profile_check.py profile.json` — valid polygon, nothing below the contact plane, flat-foot coverage, monotone top edges.
- Export STL and run `python3 /home/box/agent-data/workflows/engineering-cad-release-workflow/tools/stl_check.py part.stl --expect-len <mm> --plane-axis y --plane <foot_y> --only-abs-x <a> <b>` — watertight, volume vs plan (±0.5 %), bbox/length exact, no points below contact planes.
- API: bounding box and mass properties vs plan. Pass: all PASS. (The original bridge STL fails this check with a 1.25 mm nub — this check would have caught it.)

## Gate 6 — GUI manual triple check (see `design-concept-gate` §9)

Checklist, screenshot each: - [ ] Feature list: none red/yellow, part count right. - [ ] Iso + true profile view + section view through each joint. - [ ] **Close-up of every contact/interface**: feet, load point, joints, re-entrant corners (look for nubs, dips, slivers, notches). - [ ] Key dimensions verified (Measure if it works; otherwise bbox API + STL bounds count as the measurement — say which). - [ ] Mass properties shown and match plan. - [ ] Adversarial: top 3 ways it fails; fix before continuing. Workarounds: `onshape-gui-fast-recipes` ("Verification" section).

## Gate 7 — Metadata

After ANY regeneration/feature update: re-check part name, material, appearance, part count, and that mass properties are non-blank. Regeneration can mint a new part ID and drop metadata.

## Gate 8 — Manufacturing / real slice

Orientation chosen for strength (`3d-print-strength-optimization`), exact rule settings, real slice mass recorded (model-only, brim stated). If over target: first reduce geometry in thick non-critical regions, then infill; never cut walls on load paths. Record the slice-vs-estimate ratio back into the calibration log for that geometry class. Pass: real slice <= limit with margin (>= 3 % preferred); user-confirmed number.

## Gate 9 — Test plan

Orientation, which way up, how it sits on supports (bearing length at each fixture in range), how to hang/apply load, load steps, what fails first and where to watch, safety (catch tray, eye protection). Capacity quoted for the **actual** fixture (re-run the model; don't extrapolate).

## Gate 10 — Release and delivery

Named version ("V2 release"), STL/3MF exported from that version, re-checked by Gate 5, files delivered with: real (or clearly flagged estimated) mass, capacity range, test steps, check results per gate. Wording: FACT only for things measured this run.

## Stop-and-fix summary

Any FAIL -> fix in CAD (not by re-wording), re-run that gate and all later gates. Changing the fixture or a limit -> back to Gate 2. Real slice disagrees with estimate by >5 % -> update Gate 3 method before the next design.

## Helper scripts

`tools/stl_check.py` (numpy only) and `tools/profile_check.py` (shapely) live in this skill's folder: `/home/box/agent-data/workflows/engineering-cad-release-workflow/tools/`. Run with `--help` for options.

## Helper file: engineering-cad-release-workflow/RETRO-BRIDGE.md

# Retrospective: 3D-printed bridge challenge (V1 lost, V2 redesign) — blameless

## Rules (as finally confirmed)

Total bridge length 228.6 mm (9 in) — confirmed late as TOTAL length, not gap. Sliced mass <= 30 g (3 walls, 20 % gyroid, 0.2 mm), SUNLU HS PETG 1.27 g/cm3, H2S 0.4 mm, Textured PEI, no glue/fasteners, one top-down Onshape Part Studio, mass hung at mid-span. Test gap: unstated -> 7.5–8 in -> recommended 8 in -> actually 7.25 in.

## Timeline

1. V1: low flat multi-part truss (2 trusses + 8 ties + loading beam), ~35 mm tall. Sliced 25.73 g. No trade study.
2. Competitor (Claude bot): one-piece king-post truss, 65 mm tall, 14 mm deep, printed flat. 27.22 g; est. 55–85 kg. Judged likely ~2–3x stronger than V1.
3. 20-point critique -> new skill `design-concept-gate`, pointers in 7 skills.
4. V2: trade study of 8 forms (DESIGN-V2.md). Winner: king-post, load eye hung 11 mm BELOW the tie, tapered rafters: model 844 N (86 kg) at 8 in. Claude's design on the same model: tie governs ~66 kg.
5. V2 build via native-feature console bundle. GUI review found: V-nub under each heel foot (1.25 mm below the contact line at x≈±101, confirmed by `tools/stl_check.py` on the original STL), V-dip at heel/rafter top edge, and a "hole" at tie centre (false alarm: origin marker).
6. Fix bundle: features regenerated -> part got new ID, lost name + PETG material -> mass property blank. Restored by hand.
7. Delivered "ready to print, ~27.1 g". User's real Bambu slice: **31.09 g — over the hard limit.** User dropped infill to 15 % -> 29.90 g (0.1 g margin).

8. Test gap changed to 7.25 in after delivery; capacity re-quoted without re-running the model.

## Root causes

| # | Failure | Root cause | Number |
|---|---|---|---|
| 1 | V1 weak form | Committed to one form; invented limits (low height, "joints" => many parts, normal look) | V1 ≈ 1/2–1/3 capacity of competitor |
| 2 | Mass over limit | Raw shell+infill estimator gave 32.46 g (close to the real 31.09). It was then multiplied by a "calibration" 0.843 taken from two *different* geometries (V1 multi-part flat truss f=0.828, Claude truss f=0.843). The factor does not transfer: V2 members are thin (2–6 mm in-plane) so 3 walls x 2 sides ≈ 2.5 mm fill most of each member — the part slices nearly solid, and the true ratio sliced/raw was ~0.96. Calibration was used to make the number smaller, with only 1.8 % "spread" claimed from 2 points. Brim/settings in the user's slice were never checked. No real slice before "ready to print". | est 27.1 g vs 31.09 g (+14.7 %) |
| 3 | Nub/dip defects | Sketch polyline offsets + fillets and bar ends overlapping at nodes produced spikes below the foot line and notches on the top edge. No automated outline check; caught only by eye. | nub 1.25 mm below foot |
| 4 | Lost name/material | Updating sketches via API re-created the part (new part ID); metadata attached to the old ID was dropped. | mass property blank |
| 5 | GUI check friction | Measure tool unreliable for the agent; view-cube "Front" showed the 12 mm thickness, not the profile (part was built on the Top plane); typed values hit viewport shortcuts when dialog lost focus; laggy page. | — |
| 6 | Load case ambiguity | Gap and "9 in total vs gap" not pinned down before sizing; gap moved 8 -> 7.25 in. | 4 gap values used |
| 7 | Opponent | Only benchmarked the competitor after losing. V2 did model Claude's governing failure (tie) and beat it — that should be step 1, not a post-mortem. | — |
| 8 | Over-claiming | "Ready to print" and "~27 g" stated as FACT while sliced mass was an untested estimate; 7.25 in capacity quoted without a run. | — |

## What went well

Trade table with 8 forms; same-model competitor comparison; hung-below-tie load (self-centring, post in tension); STL re-verified (watertight, 228.600 mm, 42 693 mm3 matches plan); named release version.

## Rules adopted (see engineering-cad-release-workflow)

Confirm the fixture first; trade study + competitor model before CAD; hard limits designed to <=90 % with a pessimistic estimate; real slice in the target slicer is a release gate; automated profile/STL checks; metadata re-check after every regeneration; never say "ready" with an unverified hard-limit metric.

## Helper file: engineering-cad-release-workflow/onshape-verification-gui-notes.md

# Onshape GUI verification notes (Grok Bot) — work in progress

Scratch doc: "scratch - verification tools practice - grokbot" (personal space, Owned by me) URL: <document URL> Screen: Computer coordinates are 1280x800 (real browser px). Screenshots from Computer lag ~1 action: take a second screenshot before concluding a step failed.

## 0. Build log of the test part (what worked)

- Create doc: Documents > "Owned by me" (left nav) > Create > Document... > name field is pre-selected; type name + Enter. (NOT in "Applied Science and Engineering" workspace.) Doc defaults to INCHES.
- Sketch: click "Top" in Features tree, click Sketch button (toolbar left). Sketch opens normal-to-plane automatically (view menu has "Orient normal to sketch on edit" checked).
- Tool keys inside a sketch work only with mouse over viewport: `r` = rectangle (it was CENTER-point rect in this UI), `d` = dimension, `Escape` exits tool. Toolbar clicks are more reliable than keys.
- Dimension: press d / click Dimension icon, click an edge (hover shows "Length: x in" bottom-right), click empty space to place, a field opens WITH TEXT SELECTED; type `100 mm` + Enter (units typed explicitly convert fine in an inch doc). Display lags one frame (shows old number, then updates).
- Black sketch lines = fully constrained, blue = under-defined (I left hole positions undimensioned; fine for practice, bad for release).
- Extrude: click Extrude icon (first icon after Sketch), click region inside plate (NOT on holes; holes stay out), triple-click Depth field, ctrl+a, type `12 mm`, Enter, then click green check (Enter alone only commits the field).
- Fillet icon is at x~273,y97 in the toolbar; x~309 is CHAMFER (wrong tool opened first). Fillet: click vertical edge, radius field `5 mm`, green check.
- Isometric view: keyboard `7` did NOT work from Computer key press; use View cube dropdown (small cube icon under the view cube, ~1255,212) > Isometric.

# 1. Measure tool — WHAT WORKED (no toolbar button needed)

- Onshape's "Measure" is NOT a toolbar tool and not in "Search tools". It is automatic: select entities in the viewport and read the BOTTOM-RIGHT STATUS BAR.
- Single entity: click it -> status bar shows e.g. "Diameter: 0.315 in (8 mm)", edge "Length", etc.
- Two entities: click first, then SHIFT-click second (hold Shift; plain/ctrl clicks for me gave nothing/other panels). WAIT ~3 s: status bar shows "Parallel axes dist: 2.762 in | Min dist: 2.447 in | Max dist: 3.113 in" (two hole cylinders; centre distance = parallel axes dist).
- Units: status-bar values follow DOCUMENT units (inches by default). Convert yourself (x25.4) or change Document units to mm (Document menu > Units). Click the small icon next to a value to change/copy.
- Select the CYLINDRICAL FACE of a hole (not the circular edge) to get axis-to-axis distance.
- Gotcha: clicking elsewhere in the status-bar icons (right side) opens Custom tables / Appearance panels - avoid; that was the "other panels" failure. Right-click in viewport with selection = context menu, not measure.
- Gotcha: Computer screenshots lag one action; wait 3 s before reading.
- Screenshot: shots/01-measure-two-holes-statusbar.webp

# 2/4. Part context menu (right-click "Part 1" in Parts list at bottom-left)

Items: Rename, Properties..., Assign material..., Edit appearance..., Copy here, Create Drawing, Export..., Create task, Hide, Isolate, Make transparent, Add comment, Zoom to selection (= zoom to fit part), Delete. - GOTCHA: menu position shifts by ~13 px between openings (Assign material was y=617 once, y=604 next; my wrong click opened Edit appearance). ALWAYS screenshot the menu before clicking. - Assign material... opens "Material" panel with tabs Library | Custom. Custom has Name, Density (lb/in^3 in an inch doc - label follows document units), Poisson, Young's modulus (psi) etc. Name is required (red until filled). Library default dropdown "- None -" => "No material assigned" (mass props then unusable / default density). - For 2.7 g/cm^3 aluminium in an inch doc enter 0.0975 lb/in^3 (or switch doc units to mm/g first so the field is g/cm^3).

# ENVIRONMENT PROBLEM (blocked rest of run)

- Onshape showed "Onshape is not connected... Attempting to reconnect" then "Poor connection..." and the UI froze (typed text/clicks ignored; page reload took ~40 s). Not verified: mass properties, section view, view cube names, part rename/regen persistence, check interference, errors panel, STL export dialog.
- Lesson: if the banner "Onshape is not connected" appears, unsaved panel edits are lost (open Material panel closed itself). Wait/reload, then redo; confirm with a fresh screenshot.

## TODO (not yet practiced)

Mass properties, Section view, view-cube named views, Part rename + regen, Check interference (Assembly only normally; Part Studio has Boolean/interference via "Check" none), Errors/red features, STL export.

# ===== SESSION 2 (stable connection, fresh tab) =====

Tip: after a freeze, open the doc URL in a NEW tab (browser_run_code newPage + goto, close old). Worked first try. For panels/dialogs, browser_snapshot + browser_run_code fill() on textboxes is MORE reliable than Computer clicks (Computer screenshots lag and the Custom-material tab click appeared to do nothing until retried via getByText('Custom').click()).

## A. Document units -> mm / g / g/cm3 (WORKS)

Hamburger (≡, left of doc title, ~103,72) > "Workspace units..." (NOT called "Document units"). Panel "Workspace units": Length default unit = Millimeter, Mass default unit = Gram, Density default unit = Gram per cubic centimeter (all custom <select> dropdowns; click, WAIT ~1 s, screenshot, then click option; a rapid second click picked Kilogram instead of Gram). Green check (top right of panel) to apply. After this the Material > Custom density label reads "Density (g/cm³)" and modulus labels read Pa. Gotcha: other defaults (acceleration, force...) stay imperial; irrelevant for mass.

## B. Custom material + Mass properties (WORKS)

- Right-click Part 1 (Parts list) > Assign material... > Custom tab: Name = "Test PETG", Density (g/cm³) = 1.27 > green check.
- Mass properties is NOT in the right-click menu. It is the LAST icon in the bottom-right corner of the viewport (tooltip "Display mass and section properties", ~1267,771). Select the part first (click Part 1 in list) — panel then shows Part 1 automatically.
- Result: Mass 59.346 g, Volume 46729.248 mm³, Surface area 11725.642 mm², CoM (-0.069, 0.026, 6) mm, inertia table. Check: 46.729 cm³ x 1.27 = 59.346 g ✓. Hand check: 48000 - 2π·4²·12 (1206.4) - fillet (1-π/4)·25·12 (64.4) = 46729.2 ✓ matches.
- Panel has "Mass: Override" checkbox (tick = manual mass; leave unticked) and a Face tab (section properties of a face).
- Shot: shots/06-mass-properties-panel.webp
- Right-side vertical icons (1270,395-483) are NOT mass/measure: they are Feature/Variable table, Configurations, Custom tables, Inspection table (FeatureScript-based custom tables = what opened "other panels" before). Click the same icon again to collapse.

## C. Measure between planar faces / vertices (WORKS, mm doc)

- Single face click -> status bar "Area: 480.000 mm²" (40x12 end face; other end 420 because of 5 mm fillet). Single vertex click -> "Point: X -50.000 mm Y 20.000 mm Z 12.000 mm" (absolute coords - handy

to verify position vs origin).
- Two planar faces: click face 1, then SHIFT-click face 2 -> "Parallel dist: 100.000 mm" (plate length). Two vertices: click, shift-click -> "Min dist: 100.000 mm" (those two corners were on the same edge).
- HIDDEN FACE TRICK: selection persists while the view changes. Select the visible face, change view (view-cube Left/arrow, or View menu > Isometric), then shift-click the other face once visible. Plate end faces are never both visible in one iso.
- View cube: click a face label (e.g. "Left") = snap to that named view (looks squarely at the 40x12 end face); arrow buttons around cube rotate 90° steps; they animate slowly (wait ~3 s). Right-mouse-drag in empty viewport = orbit (works, but lands on odd angles).
- Click empty viewport (far from plates, e.g. 1100,650) to clear selection before a new measurement.
- GOTCHA: when two Computer calls are issued in parallel the 2nd click is often dropped (cursor never moves). Issue clicks one at a time. Computer screenshots often show the previous frame; read again after a wait or use browser_take_screenshot.
- Shots: 07-measure-two-vertices-100mm, 08-measure-two-planar-faces-100mm, 09-vertex-point-xyz.

## D. View menu (small cube icon under view cube, ~1250,212) contents

Isometric / Dimetric / Trimetric / Graphics preferences / Named views... / Previous view / Zoom to fit / Zoom to selection / Zoom to window / Perspective view / (x) Orient normal to sketch on edit / Shaded with edges / Hidden edges removed / Tangent edges visible / View in high quality / Highlight boundary edges / Section view... - Part right-click menu also has "Zoom to selection".

## E. Section view / named views / Normal to / Zoom to selection (WORKS)

- Section view: first select a PLANE in the Features tree (e.g. Front), then View menu (cube icon ~1250,212) > "Section view..." (last item). Panel "Section view" opens with Section plane 1 (Front) preselected; model is cut and cut faces are hatched (hole halves visible when plane passes through hole axes: holes are at Y=0 so Front plane cuts both). Green check keeps it, X cancels (view-only, does not change the model). Shot: shots/10-section-view-front-plane.webp
- Normal to / Top view: right-click "Top" plane in tree > "View normal to" (also: New sketch, Offset plane, Zoom to selection, Hide). Result: view cube reads "Top", XY plane with X right / Y up -> shows the Sketch-1 profile (plate outline, two holes, fillet at bottom-right corner). The sketch was drawn on Top, so TOP (normal to Top plane) is the view that shows the profile; Front/Right show only the 12 mm thickness edge-on. Shot: shots/11-top-view-normal-to-top-plane.webp
- Zoom to selection: right-click Part 1 > "Zoom to selection" (fits the part to the window; worked). Menu opens UPWARD when part list is low - positions vary each time, screenshot first.
- Part right-click menu AFTER a material is assigned shows "Edit material..." instead of "Assign material..." (quick indicator material is set) and adds "Show center of mass".

## NOT COMPLETED this session (stopped by parent): part rename + regen persistence, edit sketch dimension and back, errors/red features, check interference, STL export dialog.

## Helper file: engineering-cad-release-workflow/tools/profile_check.py

```
#!/usr/bin/env python3
"""2D profile checks for a flat (extruded) part, using shapely, BEFORE building in CAD.
```

```
Input JSON: {"outline": [[x,y],...], "holes": [[[x,y],...],...],
  "contact": {"y": -1.2, "x_ranges": [[-114.3,-12],[12,114.3]]},   # flat feet
  "monotone_edges": [{"x_range": [12,100], "side": "top", "dir": "up"}]}
Checks: valid polygon, no vertex below contact y anywhere (nubs), the boundary
inside each contact x-range lies ON the contact line (flat foot), and the top
boundary is monotone in each listed range (catches V-dips / notches from
polyline fillets or bar ends at nodes). Prints area. Exit 1 on FAIL.
Usage: python3 profile_check.py profile.json"""
import sys, json, numpy as np
from shapely.geometry import Polygon, LineString
d = json.load(open(sys.argv[1])); P = Polygon(d['outline'], d.get('holes', [])); ok = P.is_valid
print('valid', P.is_valid, 'area', round(P.area, 2))
xy = np.array(P.exterior.coords); tol = d.get('tol', 0.02)
c = d.get('contact')
if c:
    below = xy[xy[:, 1] < c['y'] - tol]
    print('vertices below contact:', len(below), below[:5].round(2).tolist()); ok &= len(below) == 0
    for x0, x1 in c['x_ranges']:
        cut = P.intersection(LineString([(x0 + tol, c['y'] + tol), (x1 - tol, c['y'] + tol)]))
        cov = cut.length / (x1 - x0 - 2 * tol)
        print(f'foot [{x0},{x1}] flat-contact coverage {cov:.3f}'); ok &= cov > 0.98
for m in d.get('monotone_edges', []):
    x0, x1 = m['x_range']; xs = np.linspace(x0, x1, 400); ys = []
    for x in xs:
        s = P.intersection(LineString([(x, -1e4), (x, 1e4)]))
        b = s.bounds; ys.append(b[3] if m.get('side', 'top') == 'top' else b[1])
    dy = np.diff(ys); bad = (dy < -tol).sum() if m.get('dir', 'up') == 'up' else (dy > tol).sum()
    print(f"monotone {m}: violations {int(bad)}"); ok &= bad == 0
print('RESULT', 'PASS' if ok else 'FAIL'); sys.exit(0 if ok else 1)
```

## Helper file: engineering-cad-release-workflow/tools/stl_check.py

```
#!/usr/bin/env python3
"""STL release check, numpy only (binary or ASCII STL).
Usage: python3 stl_check.py part.stl [--plane-axis y --plane 0 --tol 0.05]
       [--only-abs-x 12 114.3] [--density 1.27] [--expect-len 228.6 --len-axis x]
Reports: triangles, bbox/size, watertight (every edge shared by exactly 2 faces
with consistent orientation), signed volume, solid mass, and points below a
contact plane (e.g. a foot underside that must be flat). Exit 1 on any FAIL."""
import sys, argparse, struct, numpy as np

def load(path):
    b = open(path, 'rb').read()
    if len(b) >= 84:
        n = struct.unpack('<I', b[80:84])[0]
        if 84 + 50 * n == len(b):
            d = np.frombuffer(b[84:], dtype=np.dtype([('n', '<3f4'), ('v', '<9f4'), ('a', '<u2')]), count=n)
            return d['v'].reshape(-1, 3, 3).astype(float)
    pts = [list(map(float, l.split()[1:4])) for l in b.decode(errors='ignore').splitlines() if
l.strip().startswith('vertex')]
    return np.array(pts).reshape(-1, 3, 3)

def main():
    p = argparse.ArgumentParser(); p.add_argument('stl')
    p.add_argument('--plane-axis', choices='xyz'); p.add_argument('--plane', type=float); p.add_argument('--
tol', type=float, default=0.05)
    p.add_argument('--only-abs-x', type=float, nargs=2, metavar=('MIN','MAX'), help='limit plane check to
MIN<=|x|<=MAX (e.g. feet only, skip a hanging eye)')
    p.add_argument('--density', type=float, default=1.27); p.add_argument('--expect-len', type=float);
p.add_argument('--len-axis', default='x', choices='xyz')
    a = p.parse_args(); T = load(a.stl); ok = True
    V = T.reshape(-1, 3); lo, hi = V.min(0), V.max(0)
    print(f"triangles {len(T)}\nbbox min {np.round(lo,3)} max {np.round(hi,3)} size {np.round(hi-lo,3)}")
    q = np.round(V / 1e-4).astype(np.int64)
```

```
    _, idx = np.unique(q, axis=0, return_inverse=True); F = idx.reshape(-1, 3)
    E = np.concatenate([F[:, [0, 1]], F[:, [1, 2]], F[:, [2, 0]]])
    und = np.sort(E, 1); _, c = np.unique(und, axis=0, return_counts=True)
    _, cd = np.unique(E, axis=0, return_counts=True)
    wt = bool((c == 2).all() and (cd == 1).all())
    print(f"watertight {wt} (non-manifold/open edges {int((c != 2).sum())}, flipped/dup directed edges {int((cd
!= 1).sum())})"); ok &= wt
    vol = np.einsum('ij,ij->i', T[:, 0], np.cross(T[:, 1], T[:, 2])).sum() / 6
    print(f"volume {vol:.1f} mm3  solid mass {vol*a.density/1000:.2f} g @ {a.density} g/cm3 (NOT sliced mass)")
    if vol <= 0: print("FAIL volume <= 0 (inverted normals)"); ok = False
    if a.expect_len is not None:
        ax = 'xyz'.index(a.len_axis); L = hi[ax] - lo[ax]; good = abs(L - a.expect_len) <= 0.01
        print(f"length on {a.len_axis} {L:.3f} vs {a.expect_len} -> {'PASS' if good else 'FAIL'}"); ok &= good
    if a.plane_axis:
        ax = 'xyz'.index(a.plane_axis); sel = V if a.only_abs_x is None else V[(abs(V[:,0]) >= a.only_abs_x[0])
& (abs(V[:,0]) <= a.only_abs_x[1])]
        below = sel[sel[:, ax] < a.plane - a.tol]
        print(f"points below {a.plane_axis}={a.plane} (tol {a.tol}): {len(below)}" + (f" deepest
{below[:,ax].min():.3f} at {np.round(below[below[:,ax].argmin()],2)}" if len(below) else ""))
        ok &= len(below) == 0
    print("RESULT", "PASS" if ok else "FAIL"); sys.exit(0 if ok else 1)
main()
```

# engineering-math-proofs

**Summary (description):** Use before committing, releasing a drawing for or buying anything sized by numbers — fasteners and thread engagement, fits, bearings, shafts/columns, beams, tipping, motors/gearing/current, belts/chains/gears, CG and weight — or when validating Onshape Simulation results; produces labeled inputs, worked formulas with units and a pass/fail report.

# Engineering math proofs

> **MANDATORY FIRST STEP:** before choosing a concept or refining any design, run the `design-concept-gate` skill: list only the real constraints, challenge every self-imposed limit (height, part count, look, load case, print orientation), hand-calc 3-6 fundamentally different forms, and pick on numbers. No CAD or sizing until its decision table exists.
>
> **Native features only.** Results go into Onshape as Variables and dimensions typed by hand. Never use FeatureScript, custom features (belt/chain generators, `#Belt…` functions), Feature Studios or the FS API endpoint to compute them.

## Workflow

1. **List the claims.** Write each spec value as a check with a limit, e.g. "arm hold current < 40 A breaker". One claim per row.
2. **Label every input:**
3. **FACT:** a datasheet, standard or CAD measurement, with its source.
4. **INFERENCE:** derived, or an engineering rule.
5. **GUESS:** an assumption that must be measured. It goes on the measure list.
6. **Keep units consistent.** Work in either in–lbf–psi or mm–N–MPa. Write units in every row.
7. **Build a small library.** One pure function per formula, docstring naming the formula, no I/O.
8. **Test the library** (pytest):
9. each worked example below;
10. a limit or symmetry case;
11. a regression for every bug found.
12. **Runner report.** Each row: check | value | limit | margin | PASS / FAIL / OPEN | basis. Then the failing and open list, then specific fixes (part, feature, new value). Add a FIX row next to each FAIL.
13. **Cross-check:**
14. against Onshape Mass properties, Measure and Simulation, and any other calculation;
15. list the discrepancies rather than quietly picking one.
16. **Feed CAD.** Put results into Variables (`#arm_belt_cc = 5.186 in`) and re-run after every design change. The report is generated, never hand-edited.

## Worked formulas (each with units and a pass rule)

### Fasteners

- **Thread engagement in a tapped soft material**
- Le = As·Sp / (0.577·Sy·0.75·π·D). The 0.577 is the shear/yield ratio; 0.75 is the thread shear-area factor (INFERENCE).

- **10-32 alloy SHCS (As = 0.0200 in², Sp ≈ 140 ksi INFERENCE) into 6061-T6 (Sy = 40 ksi), D = 0.19 in, gives Le = 0.0200·140000/(0.577·40000·0.75·π·0.19) ≈ 0.271 in ≈ 8.7 threads.**

- **Pass:** engagement ≥ Le and ≥ 5 threads. Otherwise use a helicoil, a through-bolt with a nut, or a longer tap.
- **Tip landing window**
- nut height + 1–2 pitches ≤ tip protrusion ≤ blind depth − 1–2 pitches.
- A screw that bottoms stops clamping even with a full nut.
- **Pass:** the tip lands inside the window for the actual stack.
- Worked example (aluminium T-slot extrusion with drop-in T-nuts): T-nut thread 6.5 mm and slot floor 11.5 mm below the face give a window of 8.5–11.0 mm of tip depth below the face. Pick screw length (from the standard sizes) and washers so every tip lands inside it; shim with washers or change a counterbore depth rather than accept a bottoming screw.
- **Preload and torque**
- Fi = 0.75·As·Sp; T = K·D·Fi, with K = 0.20 dry or 0.15–0.18 lubricated.

- **10-32 alloy gives Fi = 0.75·0.0200·140000 = 2,100 lbf, so T = 0.2·0.19·2100 ≈ 80 lbf·in. A grade-5 equivalent gives 1,275 lbf and 48 lbf·in.**

- Cap the torque at what the weakest threaded member (T-nut, tapped aluminium, insert) can take, and print it on the drawing.
- **Pass:** the joint's separating or shear load < Fi·μ (μ ≈ 0.2, INFERENCE).
- **Pin or bolt bearing stress**
- σ = F/(d·t). F = 200 lbf on a Ø0.25 in pin through 0.125 in plate gives **6,400 psi**.
- **Pass:** below the plate's Sy/2: 20 ksi for 6061 (PASS); polycarbonate Sy ≈ 9 ksi gives 4.5 ksi, so FAIL; add a bushing.

### Fits (ISO 286), µm

| Size (mm) | IT6 | IT7 |
|---|---|---|
| 6–10 | 9 | 15 |
| 10–18 | 11 | 18 |
| 18–30 | 13 | 21 |
| - **Ø12 H7/g6:** hole 0/+18, shaft −6/−17, giving 6–35 µm clearance. Sliding or locating fit: **pass** if it isn't a press fit. | | |
| - **Ø12 H7/k6:** shaft +1/+12, a transition fit. **H7/p6:** +18/+29, interference 0–29 µm, a press fit. | | |
| - **3D prints and laser cut:** apply the measured process offset instead (GUESS until a test coupon is measured). | | |

### Bearings, columns, shafts

- **Static:** C0/P0 ≥ s0 (s0 ≈ 1–2 for shock, INFERENCE). Use static ratings for impacts, not dynamic.
- **Life:** L10 = $(C/P)^3 \cdot 10^6$ rev for ball bearings. C = 1,000 lbf and P = 200 lbf give **$1.25 \times 10^8$ rev**. **Pass** if L10 is far beyond season revolutions.
- **Euler buckling:** $P_{cr} = \pi^2 EI/(KL)^2$, with K = 1 for pinned-pinned.
- Ø0.5 in steel rod (E = $29 \times 10^6$ psi, $I = \pi d^4/64 = 0.00307$ in⁴), L = 12 in, gives $P_{cr} \approx$ **6,100 lbf**.
- **Pass** at SF = Pcr/P ≥ 3. At P = 500 lbf, SF = 12.2: PASS.

## Beams (tube frame members)

- **Deflection:** cantilever $FL^3/3EI$; simply supported centre load $FL^3/48EI$; uniform load $5wL^4/384EI$.
- **Stress:** σ = Mc/I. **Pass:** δ ≤ L/360 (or the function limit) and σ ≤ Sy/2.
- **Worked example:** 2×1×1/16 in 6061 tube, bending about the strong axis.
- $I = (1 \cdot 2^3 - 0.875 \cdot 1.875^3)/12 = 0.186$ in⁴.
- 24 in cantilever with 20 lbf at the tip: $\delta = 20 \cdot 24^3/(3 \cdot 10 \times 10^6 \cdot 0.186) =$ **0.050 in** ≤ 0.067 in (L/360), PASS.
- $\sigma = 20 \cdot 24 \cdot 1/0.186 =$ **2,580 psi** ≤ 20 ksi, PASS.

## Tipping and CG

- **CG:** $\bar{x} = \Sigma m_i x_i / \Sigma m_i$, and the same for y and z. Compare with Onshape Mass properties for the assembly.
- **Static push:** SF = W·d/(F·h_F), where d is the horizontal distance from CG to the tipping edge.
- W = 125 lbf, d = 12 in, push F = 60 lbf at h_F = 20 in gives SF = 1500/1200 = **1.25 < 1.5: FAIL**.
- Fix: lower the push point (bumper height) or the CG, or widen d.
- **End load on furniture or a cantilevered top:** solve the moment about the foot edge for the largest allowed load at the overhang, and print that limit on the drawing when it falls short of the spec.
- **Acceleration tip:** a_tip = g·d/h_cg. d = 12 in and h_cg = 10 in give a_tip = **1.2 g**.
- **Pass** if it is above the drive's achievable acceleration (wheel friction μ·g, ≈1.1 g on carpet, INFERENCE). Here it passes, but the margin is small.

## Motors, gearing, battery

- **Motor curve:** $\tau = \tau_s(1 - \omega/\omega_f)$; $I = I_f + (I_s - I_f) \cdot \tau/\tau_s$.
- **Motor specs to verify against current vendor data:**
- Kraken X60: 6000 rpm, 7.09 N·m, 366 A, 2 A free.
- NEO: 5676 rpm, 2.6 N·m, 105 A, 1.8 A free.
- **Drive speed:** v = ωf·r/G.
- Kraken, G = 6.75, Ø4 in wheel gives $(6000/60) \cdot 2\pi \cdot 2/6.75/12 =$ **15.5 ft/s free, ≈13.2 ft/s real** (85 %).
- **Pass:** the target is 80–90 % of free speed.
- **Arm hold:**
- τ = m·g·L·cosθ. 4 kg at 0.5 m, horizontal, gives **19.6 N·m**.
- Per motor: $\tau_m = \tau/(G \cdot \eta \cdot n)$. G = 100, η = 0.8 and n = 2 give **0.123 N·m**, so $I = 2 + 364 \cdot 0.123/7.09 \approx$ **8.3 A** per motor.
- **Pass:** < 40 A breaker continuous, and the motor isn't stalled long.
- **Battery sag:** V = Vb − I·R, with R ≈ 0.011–0.02 Ω (INFERENCE).
- 12.5 V − 200 A·0.015 Ω = **9.5 V**.
- **Pass:** above brownout, ≈6.8 V on roboRIO 1 and 6.75 V on roboRIO 2 (verify), with margin.

## Belts, chains, gears (replace generator features)

- **Belt:** $L = 2C + \pi(D1 + D2)/2 + (D2 - D1)^2/(4C)$, with PD = N·p/π.
- 80T 5 mm HTD (p = 0.19685 in) on 18T/36T: PD = 1.128 and 2.256 in. Solving gives **C ≈ 5.186 in**.
- WCP-style offsets (verify): gearboxes 0 to −0.003 in; rollers −0.015 to −0.020 in.
- **Chain:** $L/p = 2C/p + (N1 + N2)/2 + ((N2 - N1)/2\pi)^2 \cdot p/C$. Round to an **even** link count. PD = p/sin(180°/N).

- **25 (p = 0.25 in), 15T/30T at C = 6 in gives 70.7 links. Pick 72, then re-solve: C ≈ 6.159 in. PD(15T) = 1.202 in.**

- **Gears:** C = (N1 + N2)/(2·DP) or m·(N1 + N2)/2.
- 20 DP 14T/60T gives **1.850 in**; module 1, 20T/40T gives **30 mm**.
- **Pass:** C matches the CAD hole spacing within the process tolerance.

### Cable lengths

- Route length = sum of straight runs + bend arcs (π·r·θ/180 per bend) + service loops, plus the full travel of any moving joint (lift stroke). **Pass:** the purchased cable length ≥ route length + 10 % slack, and every bend radius ≥ the cable's minimum.

### Sheet metal flat length

- BA = θ·(R + K·T); OSSB = (R + T)·tan(θ/2); BD = 2·OSSB − BA.
- T = R = 0.090 in, K = 0.45, 90°: BA = 0.2050, BD = 0.1550. Outside legs 1 and 2 in give flat = **2.845 in**.
- **Pass:** within ±0.002 in of the Onshape flat pattern.

## Simulation cross-check

- **Hand-calculate every FEA result:** beam bending, bolt shear, buckling.
- **Simulation setup:** a fixed instance; materials on everything; only supported mates; avoid bonding all touching faces; ≤100 instances. Plan availability UNVERIFIED.
- **Modal:** keep natural frequencies away from excitation frequencies.
- **Pass:** FEA and hand calculation agree within about 20 % (INFERENCE). Otherwise explain the difference before trusting either.

## Do

- State whether a rating is static or dynamic.
- Include self-weight and the worst payload position.
- Solve the statics instead of dividing by arbitrary factors.
- Put every GUESS on the measure list and re-run once it is measured.

## Don't

- Don't round a 0.99 margin up to a pass.
- Don't mix unit systems inside a formula.
- Don't use a generator feature or FeatureScript to get a length or center distance.
- Don't quote motor or brownout values without checking the current vendor and WPILib docs.

## Common mistakes

- **Point load vs uniform load,** or the wrong span or supports.
- **Assuming "longer screw = more engagement"** past the blind-hole floor.
- **Odd chain link counts,** which need a half link.
- **Using pitch diameter vs outside diameter** wrongly in belt length.
- **Free speed used as real speed;** stall current ignored for breakers.

## Checks before done

- [ ] Every input is labeled FACT, INFERENCE or GUESS, with a source.
- [ ] Units are in every row; tests pass and the count is reported.
- [ ] Every check shows value, limit, margin and PASS/FAIL/OPEN.
- [ ] FAIL rows have specific fixes that were re-checked.
- [ ] Results are entered as Onshape Variables or dimensions, and the CAD is re-measured.
- [ ] Discrepancies with Onshape Mass properties, Measure or Simulation are listed.

## Report format

```
| # | Check | Value | Limit | Margin | Result | Basis |
| 1 | #10-32 engagement in 6061 | 0.30 in | ≥0.271 in | +11 % | PASS | FACT As; INFERENCE Sp |
| 2 | Push tip SF | 1.25 | ≥1.5 | −17 % | FAIL | GUESS push height |
| 2F | FIX: bumper push at 12 in | 2.08 | ≥1.5 | +39 % | PASS | |
Measure list: push height, battery R, print offset
```

## Sources

- Onshape Simulation: https://cad.onshape.com/help/Content/Assembly/simulation.htm , https://cad.onshape.com/help/Content/Assembly/modal_simulation.htm , https://cad.onshape.com/help/Content/Assembly/asynchronous_simulation.htm
- Mass properties and Measure: https://cad.onshape.com/help/Content/View/mass_properties_tool.htm , https://cad.onshape.com/help/Content/View/measure_tool.htm
- UNSOURCED, from standard machine-design practice (Shigley-type textbook formulas): thread engagement (the 0.75 factor is INFERENCE), preload/torque K, Euler, beam formulas, L10, belt and chain length, bend allowance.
- ISO 286 values are from memory of the standard (the standard is paywalled; verify).
- Motor, brownout, battery R and WCP offsets: verify against current vendor and WPILib documentation.

# custom-part-design-from-requirements

> **Summary (description):** Use when an FRC (or other) mechanism needs a NEW custom part designed in Onshape rather than a COTS part mated in (plates, gussets, bearing blocks, collars, motor plates, pocketed arms, tensioners, hooks, bent brackets, printed mounts), to go from requirements (loads, interfaces, process, mass) to a fully-defined, manufacturable Onshape Part Studio with checked mass, walls, corner radii and hole sizes.

# Custom part design from requirements (Onshape-only)

> **MANDATORY FIRST STEP:** before choosing a concept or refining any design, run the `design-concept-gate` skill: list only the real constraints, challenge every self-imposed limit (height, part count, look, load case, print orientation), hand-calc 3-6 fundamentally different forms, and pick on numbers. No CAD or sizing until its decision table exists.

Labels: FACT = sourced (see /workspace/skill-drafts-parts/custom-part-design-from-requirements/SOURCES.md), INFERENCE = derived from facts, GUESS = assumption to confirm. Every part is modelled in Onshape with built-in features only (Sketch, Extrude, Revolve, Fillet, Chamfer, Hole, Shell, Pattern, Mirror, Sheet metal). Never FeatureScript. Other CAD tools only for a scratch numeric cross-check. Load path and printed-part strength: `load-path-simulation-for-assemblies`, `3d-print-strength-optimization`.

## 1. Process (in order; each step produces something checkable)

1. **Requirement card (5 lines before any sketch).** Function; interfaces (which COTS part, which hole pattern, drawing URL); load case (N or lbf, where, direction); process (router plate, laser + bend, lathe, mill, FDM); mass budget.
2. **Pull interface numbers from the vendor drawing, not memory.** Example, Kraken X60: 11× #10-32 on a 2.000 in bolt circle at 30° spacing, 0.750 in pilot, blind 0.25 in holes (FACT, WCP drawing/docs). Put them in Variables (`#bc = 2 in`, `#pilot = 0.75 in`).
3. **Pick the process early; it sets the rules.**
4. Router plate: every inside corner radius ≥ tool radius + 0.02 in, or ~1.3× tool radius (FACT). 1/8 in cutter → inside R ≥ 0.085 in; 1/4 in cutter → ≥ 0.145 in (INFERENCE). Ask which cutter the team uses (GUESS until asked).
5. Laser cut + bend, 0.125 in 5052, SendCutSend (FACT): effective bend R 0.125 in; min flange 0.476 in after the bend, 0.368 in in the flat; max bend length 44 in; one vendor lists K = 0.44, BD 0.216 in. Onshape default K is 0.45 (FACT); match the vendor.
6. Tapped holes (FACT): #10-32 tap #21 0.159 in; #10 clearance #9 0.196 (close) / #7 0.201 (free). #8-32 tap #29 0.136 in; #8 clearance #18 0.1695 (close) / #16 0.177 (free).
7. **Layout sketch first.** Place interface features (bores, bolt circles, tube holes) as construction geometry dimensioned to the origin or a datum, then draw the outline around them. Wall rule (INFERENCE from the drills): edge distance ≥ 2× hole diameter for #10 (~0.4 in). Measure walls from the largest feature.
8. **Feature order.** Profile → Extrude NEW; interface cuts (Hole feature for threaded/counterbored; Extrude Remove for bores and slots); lightening pockets; edge fillets and chamfers. Name every feature. Drive thickness and width with `#t` / `#W`.
9. **Assign material and part name/number**, check Mass properties, and compare with a hand estimate (area × t × ρ). Densities g/cm³: 6061 2.70, 7075 2.81, 5052 2.68, POM 1.41, PETG 1.27.
10. **Checks before release.** Minimum web and edge distance (Measure); minimum inside radius vs cutter; hole sizes vs chart; interference in the assembly (right-click → Check interference); one-line stress hand

calc (§3).

11. **Version** ("V1 for review"), then drawing or DXF export (right-click face → Export as DXF) or the Sheet metal flat pattern. See `onshape-professional-drawings`, `onshape-release-and-versioning`.

# 2. Archetype recipes (reference parts in Onshape doc "Custom Part Practice Drills": <document URL>, tabs D01–D11)

| Archetype | Onshape feature strategy | Key numbers |
|---|---|---|
| Spacer (D01) | Top sketch: 2 circles → Extrude NEW `#L` → Extrude Remove bore through | OD 0.5, bore 0.201 (#10 free), POM |
| Clamp collar (D02) | Circle → Extrude `#W`; bore + 0.04 slit cut through; Right plane: one **Hole** feature (Counterbored, Tapped 8-32) across the slit | Reference used 3 cuts with a start offset; the Hole feature is cleaner |
| L gusset (D03) | Polygon with 1.25 in chamfers, 0.25 in inner fillet → Extrude `#t`; 10× #10 free on a 1 in grid; triangular pockets r 0.09 | 0.125 6061, 69.7 g |
| Bearing plate (D04) | Rounded rectangle → bore 1.123 (FR8 press, 1.125 OD bearing) + 4× 0.201 | Edge distance 0.399 in |
| Bearing block (D05) | **Front-plane** side profile (shoulders + arc cap) → **symmetric** Extrude `#W`; bore symmetric; tap holes from Top | Thinnest wall over the bore 0.187 in |
| Motor plate (D06) | Rounded plate; pilot slot 0.770 wide with 0.125 travel (tensioning); 6× 0.201 **slots** on r 1.0 | Web between slots 0.408 in |
| Bent bracket (D07) | **Sheet metal model** (t 0.125, R 0.125, K = vendor) → Flange | Flat ≈ 1.25 + 1.0 + BA 0.283 = 2.53 in (K 0.44) |
| Pocketed arm (D08) | Tangent hull of 2 circles → Extrude; bore + tip hole; Warren-truss pockets (rim 0.2, web 0.2, fillet 0.15) | 121.9 g at 0.25 in 6061 |
| Tensioner (D10) | Filleted polygon; pivot hole; idler hole; **arc slot** R 1.25, 25–60° | Slot end r 0.1005, fine with 1/8 in cutter |
| Climber hook (D11) | Ri 0.9 / Ro 1.9 arc + shank to y = 6, filleted tip → Extrude; 2× #10 holes | 7075, 125.4 g |
| Printed mount (D09) | Tilted plane (Plane → Line angle) → sketch → Extrude Add; Shell walls; Fillet ≥ 1 mm inside | PETG; see `3d-print-strength-optimization` for orientation |

# 3. Worked numeric examples

- **Arm bending (D08).** 50 lbf tip (GUESS) at 10 in → M = 500 in·lbf. Solid root 0.25 × 2.0: $S = bh^2/6 = 0.167$ in³ → σ ≈ 3.0 ksi. Through the first pocket (two 0.2 in flanges): I ≈ 0.081 in⁴, c = 1.0 → σ ≈ 6 ksi. 6061-T6 yield ~40 ksi → ~6× margin (INFERENCE). Thickness is the main weight lever; small pockets save little.
- **Hook (D11), curved beam (Winkler).** Ri 0.9, Ro 1.9, h 1.0, t 0.25: rn = h/ln(Ro/Ri) = 1.338; rc = 1.4; e = 0.062. At 150 lbf (125 lb robot × 1.2, GUESS): inner-fibre σ ≈ M(rn−Ri)/(A·e·Ri) + W/A ≈ 6.6 + 0.6 = 7.2 ksi; at 2× shock ≈ 14.4 ksi. 7075-T6 yield ~73 ksi (INFERENCE, handbook).
- **Bend flat length (D07).** BA = π/2·(R + K·t) = π/2·(0.125 + 0.44·0.125) = 0.283 in → flat 2.53 in. In Onshape, read it from the **Flat** view.
- **Mass check (D04).** 3.5 × 2 × 0.25 in minus holes → 23,880 mm³ × 2.70 = 64.5 g; Onshape reported 23,879.79 mm³ (FACT, this build).

# 4. Lessons from building the drills natively

1. **Extrude start offset follows the sketch normal, not the flipped extrude direction.** With "opposite direction" on, also flip the start-offset arrow. D02 was 3.2 % light until fixed. Always compare Onshape mass with a hand estimate.
2. **One profile sketch per body, separate cut sketches.** Selecting the whole sketch selects every region, so holes drawn inside a profile extrude as solid slugs unless you pick regions.
3. **Symmetric extrude from a mid-plane** (D05, D07) keeps features from other planes centred and `#W` changes both sides.

4. **Slots, not round holes, where you need adjustment** (motor plate, tensioner): one Slot entity in the GUI, free belt/chain tension.
5. **Check walls from the largest feature** (D05 bore-to-cap wall 0.187 in is the real limit).

## 5. Open questions to ask the team

Router cutter diameter; bend vendor and K-factor; climber bar OD and robot weight; camera hole pattern for D09; whether drawings should carry Hole-feature thread callouts (recommended).

# cad-part-modeling-practice-drills

> **Summary (description):** Use when the user, a student, or I want to practise designing NEW custom parts in Onshape (not mating COTS parts): a graded set of 15 timed drills with spec, target time, native-feature strategy and pass criteria, usable as a 30-minute self-test, with reference answers in the Onshape doc "Custom Part Practice Drills".

# Onshape custom-part practice drills

**Reference doc:** <document URL> - One Part Studio per drill, D01–D11 (D09 is GUI-only, not built). Version: "V1 Drills D01-D11 built". - References use only Variable, Sketch and Extrude so the feature list is easy to read. GUI answers should use Hole, Slot, Fillet and Sheet metal where noted. - Process and design rules: `custom-part-design-from-requirements`.

**Rules:** Onshape only, built-in features only, never FeatureScript. Make a NEW doc for attempts; don't edit the reference doc (branch or copy it).

**Self-test (30 min):** 1 Bronze, 1 Silver and 1 Gold back to back. Pass = every criterion met within 1.5× target time. Score = drills passed / 3. Record the result and time per drill.

**Universal pass criteria:** (a) sketches fully defined (all black); (b) key sizes from named Variables (`#t`, `#W`…); (c) material set and mass within ±1 % of reference; (d) every feature named; (e) changing the main variable rebuilds with no red features.

**Checks:** Mass properties panel; Measure between edges for walls; Measure on arcs for inside radius; insert into an Assembly with the mating COTS part → Check interference.

## Bronze (target 4–6 min)

| # | Spec | Feature strategy | Extra pass checks | Ref mass |
|---|---|---|---|---|
| D01 Round spacer | OD 0.5, Ø0.201 bore, `#L` 0.5, Delrin | Top sketch circle → Extrude #L; circle → Extrude Remove through (or Hole) | Change L to 0.25 and it rebuilds | 1.9 g (1348.8 mm³) |
| D04 Bearing plate | 3.5×2×0.25, R0.25 corners, Ø1.123 FR8 press bore, 4× #10 free at (±1.25, ±0.5) | Center rectangle + sketch fillet; Hole feature (Simple, #10) | Edge distance ≥ 0.39 in | 64.5 g |
| D03 L gusset | 4.5×4.5×0.125 L for 2×1 tube, 1.25 chamfers, 0.25 inner fillet, 10× #10 free on 0.5/1.0 grid, 2 triangular pockets (r 0.09) | Polygon → Extrude; Hole on sketch points; Linear pattern | Pocket corner r ≥ 0.085 (1/8 in tool); pocket-to-hole web ≥ 0.25 | 69.7 g |
| D12 Tube end plug* | Revolve: 1.75×0.75 tube ID plug with 0.1 in lip, #10-32 tapped centre | Front half-profile → Revolve 360; Hole (Tapped 10-32, #21 0.159) | Hole callout reads 10-32 | not built |

## Silver (target 8–10 min)

| # | Spec | Feature strategy | Extra pass checks | Ref mass |
|---|---|---|---|---|
| D02 Clamp collar | OD 1.25, `#W` 0.375, 1/2 in round bore, 0.04 slit, 8-32 clamp screw across the slit | Extrude disc; cut bore + slit; Right plane: ONE Hole (Counterbored, Tapped 8-32 #29 far side, #18 clearance near side) | Counterbore on clearance side; screw crosses slit; ref volume 5747.1 mm³ | 15.5 g |
| D05 Bearing block | Front side profile: 2.5 base, 0.5 shoulders, R0.749 cap about (0, 0.75); symmetric `#W` | Front sketch → Extrude symmetric; bore symmetric; Hole from Top, blind 0.4 | Wall over bore ≥ 0.18 | 49.3 g |

| # | Spec | Feature strategy | Extra pass checks | Ref mass |
|---|---|---|---|---|
| | 0.75; FR8 bore; 2× 10-32 tapped from bottom at x ±1.0 | | | |
| D06 Kraken motor plate | 3.25×4.375×0.25 rounded; 0.770 pilot slot with 0.125 travel; 6× #10 slots on 2.000 BC at 60°; 3 tube holes | Slot tool; Circular pattern ×6; BC from the WCP drawing | Slot web ≥ 0.4; motor face clears in assembly | 145.6 g |
| D07 Bent L bracket | 5052 0.125, 1.25 base, 1.0 wall, 1.5 wide, inner R 0.125, 2 holes per leg | Sheet metal model (t 0.125, R 0.125, K 0.44) → Flange; Hole; check Flat view | Flange ≥ 0.476 after bend; flat ≈ 2.53 in | 20.3 g |
| D10 Tensioner arm | 0.1875 plate; pivot #10 at (0,0); idler 1/4 at (2,0); arc slot R1.25, 25–60°, #10 wide | Arc slot = 2 concentric arcs + tangent end arcs (or Slot on a construction arc) | Slot end r = 0.1005; rotates without hitting idler | 38.3 g |

## Gold (target 12–15 min)

| # | Spec | Feature strategy | Extra pass checks | Ref mass |
|---|---|---|---|---|
| D08 Pocketed arm | 10 in centres, R1.0 / R0.625 tangent hull, 0.25 6061; FR8 bore + 3/8 tip; Warren-truss pockets, rim 0.2, web 0.2, fillet 0.15 | Hull sketch (tangent lines); pocket sketch with offset rim; sketch fillets | Every web ≥ 0.2; mass ≤ 125 g; σ < 10 ksi at 50 lbf | 121.9 g |
| D11 Climber hook | 7075 0.25; Ri 0.9 / Ro 1.9 throat; shank to y = 6; filleted tip; 2× #10 | Arcs + lines fully defined; sketch fillet; Hole | Throat fits the bar (OD GUESS, ask); Winkler σ < 15 ksi at 2× load | 125.4 g |
| D09 Camera mount* | PETG, 20° tilted face, 4× M2 holes, Shell 2.5 mm, base 2× #10 | Plane (Line angle 20°) → sketch → Extrude Add; Shell; Fillet ≥ 1 mm inside | No wall < 2 mm; prints without supports | not built |
| D13 Gearbox side plate* | 0.25 plate, 2 FR8 bores at 14T/50T 20DP centre distance 1.600 + 0.003 in, 4 standoffs | Variable `#cd = (14+50)/(2*20) in + 0.003 in` | Centre distance driven by variable | not built |
| D14 Configured spacer family* | D01 with Configuration L = 0.125/0.25/0.5/1.0 | Configuration panel → List; per-config part number | 4 configs rebuild | not built |
| D15 In-context bracket* | Bracket whose holes follow a tube in an assembly | Edit in context; Use (project) edges | Holes stay aligned after the tube moves | not built |
| * = spec only, build in the GUI. | | | | |

## GUI-only steps (not in the API references)

Hole callouts/threads, 3D edge fillets, Sheet metal flat pattern, tilted planes, configurations, in-context references.

## Grading notes / lessons

- **Start offset direction:** with a start offset AND "opposite direction", flip the offset arrow too, or the cut lands on the wrong side (D02 reference was 3.2 % off until fixed).
- **One profile sketch, separate cut sketches:** picking the whole sketch picks every region.
- **Watch the clock:** most time is lost on under-defined sketches. Dimension to the origin first, then to features.
- **Check mass against a hand estimate** (area × t × ρ) every time; it catches a wrong extrude direction or missing cut instantly.

# Onshape CAD

# onshape-api-rate-limits

> **Summary (description):** Use when planning, writing or running anything that calls the Onshape REST API (browser-console bundles using the signed-in session, API keys, or OAuth), when a job will make more than ~50 calls, after any 429/503/402 from Onshape, or when choosing between the API, the GUI, or several agents for a bulk CAD build. Covers call budgets, estimating, batching, dry runs, stopping on 429, resuming, GUI fallback, and avoiding Onshape lag.

# Onshape API rate limits and fast builds

Goal: **never hit a 429.** If one happens anyway, stop right away and continue in the GUI or in a later run. Native features and REST endpoints only. **Never use FeatureScript** (no custom features, no evaluateFeatureScript).

## Facts (recheck if they look old)

- A 429 is **per endpoint**. The per-window limit is not published. Responses carry `X-Rate-Limit-Remaining`; a 429 carries `Retry-After` in seconds. Source: https://onshape-public.github.io/docs/api-adv/errors/
- **Annual** quotas (Enterprise 10k/user, Pro 5k, EDU/Free 2.5k; going over returns 402) apply to API keys and OAuth, not to browser-session calls. 4xx and 5xx responses don't count. Source: https://onshape-public.github.io/docs/auth/limits/
- Measured (inference): browser-session REST calls ARE throttled per endpoint. About 600 calls in 30 min gave a 429 on GET /features with Retry-After of about 56 min. About 210 calls in 10 min across two runs were fine.

## Budget policy (defaults)

| Cap | Value |
|---|---|
| Pace | 20 calls/min, at least 3 s apart |
| Rolling hour, all endpoints (ledger in localStorage: timestamps and endpoint family only) | 200 |
| Rolling hour, one endpoint family (all feature POSTs count as one family) | 150 |
| Per run | about 120-150, and never more than the hour has left |
| After any 429 | hour cap 100 for 3 h; no new run until the stored cooldown ends |
| X-Rate-Limit-Remaining | at or below 50 (or 25% of max seen), slow down 3x; at or below 10 (or 5%), stop |

## Rules

1. **Estimate before you send.** Each unit (a part studio, a phase) works out its call count first and doesn't start unless it fits the run, hour and family budgets left. Stop between units, never mid-unit.
2. **Lean scripts, not all-in-one bundles.** Per part: 1 features GET, deletes of old features, 1 POST per feature (check featureState in the response, with no extra reads), 1 mass-properties GET (volume check plus part ids), and 1 metadata POST (name plus material). A part with about 10 features costs about 15 calls.
3. **Batch:** `assemblies/.../transformedinstances` inserts many instances, each at its own transform. `.../modify` takes transformDefinitions, deleteInstances, editDescription. Read each element once and cache it. No insert-then-move per instance, no before-and-after reads, no retry loops.

4. **Dry-run first** for new jobs or anything over 100 calls: print the plan and send nothing.
5. **One phase per paste,** and at most one big phase per rolling hour (parts, then placement, then fasteners and mates).
6. **On 429:** if Retry-After is 60 s or less, wait and retry at most twice with slower pacing. If it's over 60 s or missing, stop at once and store a cooldown. For 503/500, at most 3 tries.
7. **Resume, don't redo.** Runs are idempotent (they check existing items by name or volume). Skip parts that already have working hand-built features unless forced. Make one document version before deleting anything.
8. **Known API gaps:** Hole features posted by API often fail, so script holes as extrude cuts and add real Hole features in the GUI where thread callouts or the drawing need them. Fixing an instance, some groups and relations, and inserting Standard Content seeds are GUI steps.
9. **The GUI is the backup, so make it fast:** write a click-level recipe per part up front (plane, exact dimensions, regions, feature options, verify step). Measured GUI speed was about 20 min per part, versus about 1 min per part with the lean script.
10. **Avoid lag:** never put several agents or browsers in the same document at once (it caused freezes and 10-15 s per action). Split work by document or run one at a time. Keep the browser on a good network (route through the user's own computer when it's online). Close DevTools when you're clicking in the GUI. Wait out 'Preparing' spinners instead of reloading.
11. **Console hygiene:** clear the console input (Ctrl+A, Backspace) before every paste or line. Reload the page after a run so new tabs show.
12. **Security:** never log, print or store cookies, XSRF tokens, other tokens or passwords. Send XSRF only on non-GET requests. Write only to documents the user said may be written.

## Checklist before pasting a script

- [ ] The dry-run total fits the hour and family budgets left.
- [ ] No active cooldown.
- [ ] One phase selected.
- [ ] Console input empty.
- [ ] No other agent editing the document.
- [ ] Afterwards: read the calls used, the rate-remaining header and the stop message, and update the estimates.

# onshape-assemblies-and-mates

> **Summary (description):** Use when building, organizing or reviewing an Onshape Assembly — inserting parts, choosing mates and relations, placing fasteners with Standard content and Replicate, organizing instance folders, mirroring, display states, named positions, exploded views, BOM and interference checks.

# Onshape assemblies and mates

> **Native features only.** Never use FeatureScript: no custom features, Feature Studios, `/featurescript` or evalFeaturescript. Assemble with the built-in Insert, mates, relations, Replicate, patterns, Mirror and Standard content, the way a person would.

## When to use

- A new assembly or subassembly, or a review of someone else's.
- Motion isn't behaving, parts float, or the solve is slow.
- Fastener placement, mirroring a side, or preparing BOM and drawing views.

## Workflow

1. **Plan the tree to match the physical build** before inserting anything:
2. top assembly → subassemblies (Drivetrain, Intake, Arm, Elevator, Electronics) → parts and hardware;
3. tools: "Create empty subassembly", "Move to new subassembly", "Dissolve subassembly".
4. Name the assembly tabs with the team scheme. Postman uses `(…-A100) Drivetrain`, `(…-A200) Arm`, `(…-A400) Electronics`; see `frc-robot-cad-organization`.
5. **Insert (i):**
6. From other documents, always insert from a **version**. Same-document Part Studios can be inserted from the workspace.
7. Insert configured parts and pick the configuration in the dialog.
8. **Fix one ground instance** (Shift+j), e.g. the frame or the master-layout mate connector part. Everything else is mated.
9. **Mate each pair once, choosing the type that matches the real DOF:** | Mate | DOF | Typical robot use | |---|---|---| | Fastened | 0 | Bolted plates, bearings in bores, motors on plates | | Revolute | 1 rotation | Shafts, pivots, wheels, rollers | | Slider | 1 translation | Linear slides, elevator stages | | Cylindrical | 1 rotation + 1 translation | Shaft that can also slide; spacers on shafts | | Planar | 3 | Parts sliding on a plane (rare) | | Ball | 3 rotations | Ball joints, rod ends | | Pin slot / Parallel / Tangent / Width | special | Cam slots, centering, belt contact |
10. **Couple the motion with relations:**
11. Gear (gear pairs, sprockets/pulleys at a ratio);
12. Rack and pinion;
13. Screw (leadscrews);
14. Linear (stage-to-stage elevator ratios).
15. Postman's arm assembly uses Revolute, Cylindrical and Fastened mates plus Gear relations.
16. **Pick mate connectors:**
17. Implicit points (hole centers, circular edges, centroids, midpoints) are fine when the geometry is stable. Hold Shift to lock the inference face.
18. Use **explicit Mate connectors** (in the Part Studio or the layout) where a later edit could move an implicit point. Jackhammer has explicit mate connector features in its Part Studios.

19. **Name the mates by function:** `Drive`, `6201 to Gear`, `Arm Pivot`, `Intake Roller L`. Jackhammer names its key mates this way, and it makes review and the Mates list readable.
20. **Fasteners:**
21. Insert → Standard content (ANSI/ISO/DIN bolts, nuts, washers, PEM). Use **Auto-size** on a Hole-feature hole.
22. Mate one seed fastener, then **Replicate** it onto all matching holes. Don't Group the seed; replicating groups isn't supported.
23. COTS (motors, bearings, gears) come from vendor models or FRCDesignLib (MKCad is deprecated per https://frcdesign.org/resources/frcdesignlib/ — see `frc-robot-cad-organization`). Postman uses VEX standard content for its COTS.
24. **Organize the Instances list with folders by function:**
25. Jackhammer's top assembly uses `DRIVETRAIN`, `GEARS`, `HARDWARE`, `POWER`, `PATTERNS`, `BEARINGS`, `ACCESSORIES`, `SEALS`.
26. Postman's drivetrain uses `#8-32 Nuts`, `Battery/Radio`.
27. **Patterns and mirror:**
    - Use assembly Linear/Circular patterns for repeated parts (not Part Studio patterns).
    - **Mirror** only fully constrained subassemblies, and review the per-instance action (reuse vs. mirrored part).
28. **Keep the top assembly simple:**
    - Postman's top assembly has 5 instances and only 4 Fastened mates: each subassembly is fastened to the frame or layout, with the motion inside the subassemblies.
    - Top-level configurations swap alternates by suppressing instances (Wheel Type).
29. **Presentation states:**
    - Display states (named show/hide sets), Named positions (stow/intake/score), Exploded views for drawings. You can't add mates in Explode mode.
30. **Verify:**
    - Drag-test every DOF; use Show mates mode to see the mates.
    - Run Interference detection; check that the BOM is complete (add non-geometric items like loctite, zip ties, paint).

# FRC mate recipes (INFERENCE, built from the native tools above)

| Mechanism | Mates and relations |
|---|---|
| Wheel on a dead or live axle | Revolute (wheel to tube), Gear relation from the motor pinion to the wheel at the total ratio |
| Belt or chain reduction | Revolute on each pulley; Gear relation with ratio = driven teeth / driver teeth; belt modeled as a native part, Fastened or hidden in motion studies |
| Arm pivot | Revolute at the layout `MC Arm Pivot`; a mate limit only if the travel range must be checked |
| Elevator (continuous rigging) | Slider per stage; Linear relation between stages (1:1 for continuous, 2:1 for cascade, by geometry) |
| Leadscrew stage | Cylindrical for the screw, Slider for the nut carriage, Screw relation with the lead |
| Swerve module | Revolute for the steer, Revolute for the wheel; Gear relations from each motor |
| Intake roller | Revolute; a Gear relation if it is belt-linked to other rollers |
| - Check every ratio against the hand calculation in `engineering-math-proofs` before trusting the animation. | |

# Do

- One ground per assembly; everything else mated.
- Fully constrain subassemblies that don't move, so they are rigid and faster.
- Use explicit mate connectors from the layout for subassembly placement.

- Use Replicate + Standard content Auto-size for hardware.
- Name the mates, folders and subassemblies.
- Use relations instead of driving two mates by hand.

## Don't

- Don't Group instances from different Part Studios. Group ignores geometry changes, so parts overlap after edits.
- Don't drive Animate with Fastened, Tangent or Ball mates.
- Don't pile on mate limits; they slow the solve. Use a few, only where motion range matters.
- Don't leave floating parts. Fix, mate or lock every instance.
- Don't insert cross-document parts from workspaces.
- Don't use custom features to make fasteners, belts or chains. Model the belt or chain as a native part (sketch + extrude/sweep) at the hand-computed center distance.

## Common mistakes

- **Mating everything Fastened to the ground** "for now". Real motion then can't be checked, and the solve gets slower. Use the true DOF.
- **Duplicate or redundant mates** between the same pair, which over-constrain and make the solve flip. One mate per pair.
- **Implicit connectors on features that get edited.** Mates jump after an edit. Use explicit connectors.
- **Mirroring a partially constrained subassembly.** The result is wrong. Constrain it first.
- **Hardware inserted one by one.** It is slow and inconsistent. Seed + Replicate.
- **The BOM is missing items.** Add non-geometric items, and set materials so the weight is right.

## Checks before done

- [ ] The tree matches the build; tabs, subassemblies, mates and instance folders are named.
- [ ] Exactly one fixed instance per assembly; no floating parts (drag test).
- [ ] Mate types match the real DOF; relations exist for coupled motion.
- [ ] Cross-document inserts come from versions.
- [ ] Fasteners come from Standard content, placed with Replicate.
- [ ] Interference detection is clean (or the interferences are intentional, e.g. press fits, and documented).
- [ ] Named positions, display states and exploded views exist where drawings need them.
- [ ] The BOM is complete, with materials and part numbers.
- [ ] No custom features anywhere in the referenced Part Studios.

## Sources

- Assembly basics: https://cad.onshape.com/help/Content/Assembly/assembly.htm ; Insert: https://cad.onshape.com/help/Content/Assembly/insert_parts_and_assemblies.htm
- Mates: https://cad.onshape.com/help/Content/Assembly/mates.htm , https://cad.onshape.com/help/Content/Assembly/fastened_mate.htm , https://cad.onshape.com/help/Content/Assembly/revolute_mate.htm , https://cad.onshape.com/help/Content/Assembly/slider_mate.htm , https://cad.onshape.com/help/Content/Assembly/cylindrical_mate.htm
- Mate connectors: https://cad.onshape.com/help/Content/Assembly/assembly_mate_connector.htm ; show mates: https://cad.onshape.com/help/Content/Assembly/show_mates_mode.htm
- Relations: https://cad.onshape.com/help/Content/Assembly/relations.htm , https://cad.onshape.com/help/Content/Assembly/gear_relation.htm ,

https://cad.onshape.com/help/Content/Assembly/screw_relation.htm , https://cad.onshape.com/help/Content/Assembly/linear_relation.htm

- Standard content: https://cad.onshape.com/help/Content/Assembly/standard_content.htm ; Replicate: https://cad.onshape.com/help/Content/Assembly/replicate.htm
- Group: https://cad.onshape.com/help/Content/Assembly/group.htm ; Mirror: https://cad.onshape.com/help/Content/Assembly/assembly_mirror.htm ; patterns: https://cad.onshape.com/help/Content/Assembly/assembly_linear_pattern.htm , https://cad.onshape.com/help/Content/Assembly/assembly_circular_pattern.htm
- Display states / named positions / exploded views: https://cad.onshape.com/help/Content/Assembly/display_states.htm , https://cad.onshape.com/help/Content/Assembly/named_positions.htm , https://cad.onshape.com/help/Content/Assembly/exploded_views.htm
- Instances list: https://cad.onshape.com/help/Content/Assembly/instances_list.htm ; Replace instance: https://cad.onshape.com/help/Content/Assembly/replace_instance.htm
- BOM: https://cad.onshape.com/help/Content/Assembly/bill_of_material.htm ; Interference: https://cad.onshape.com/help/Content/View/interference_detection.htm
- Performance (mate limits): https://cad.onshape.com/help/Content/Home/performance_panel.htm
- Onshape robotics onboarding (MKCAD, Standard content): https://www.onshape.com/en/blog/how-to-onboard-your-first-robotics-team
- Observed practice: Onshape Explore "Postman" (4-Fastened top assembly, configured alternates, Replicate, instance folders) and "Jackhammer" (function folders, named mates, explicit mate connectors).

# onshape-gui-fast-recipes

**Summary (description):** Use before driving the Onshape web GUI by clicking (box browser at 1280x800): tested button positions, shortcuts, step orders and gotchas for docs, sketches, dimensions, variables, extrude, revolve, patterns, mirror, chamfer, fillet, shell, assembly insert and mates, plus Learning Center navigation.

<!-- changelog (bridge retro, 2026-10-28): 1. New Verification section: workarounds for Measure, view cube orientation, dialog focus, lag. 2. Bbox API + STL bounds as accepted measurement fallback. 3. Close-up checklist for contact/interface features. -->

# Onshape GUI fast recipes (1280x800 screen)

GUI only. Never FeatureScript. Prefer Computer clicks at screen pixels; Onshape toolbar buttons have no accessible names, so role locators fail. Playwright page coordinates (1920x1113 viewport, ~x1.5 and +58 px y) miss targets, so don't mix systems. Tested = verified by hand; Untested = from lesson text.

## Timing

- Screenshots lag and the viewport animates 1-2 s. After a click wait 2-3 s and re-screenshot before deciding it failed (the same stale frame often comes back). Never re-click blindly.
- After sketching, press Esc before clicking accept, or the next click lands as a sketch point.
- Each click is its own Computer call; chained 'then' steps only allow move/scroll/wait.

## Typing values (tested, most important)

- Reliable: click INTO the field, ctrl+a (only once the field has focus, or it selects the whole page), type the value and unit as ONE string with no space ('20mm', '10mm'), then READ the field back before accepting.
- Don't type the number and unit as separate chunks with a space: the unit can land in the next field (e.g. 'mm' went into Instance count) or get scrambled ('m1m0').
- Return in a field accepts the WHOLE dialog. Tab moves focus to the next field. Use them knowingly.
- Fields convert units to doc units (20 mm shows 0.787 in); check the number, not the unit label.
- Click the canvas before shortcuts like shift+7, or they type into a still-focused field.
- Before any ctrl+a or Backspace, confirm the Onshape tab is active; keep only the Onshape tab open while modeling.
- A click sometimes registers only after a mouse move: move, then click.
- If keystrokes drop the first character (common right after Esc or a tool change), filling the visible field with a Playwright locator.fill('10 mm') then Tab works. That's ordinary field input, not the API or console.
- Typing digits right after a sketch click with no field focused triggers hotkeys (Trim, Loft, etc.). Only type once a value box is visibly open.
- Esc turns the active sketch tool OFF, and the next click then selects geometry instead of drawing. Re-pick the tool after Esc.
- Sketch on a face: click Sketch once, then click the face once. A second Sketch click resets the selection.
- Computer screenshots can be stale; browser_take_screenshot was often more current.

## Revolve (tested)

Sketch on Front: profile + construction centerline (q toggles construction). Press Esc to leave the sketch tool, then green check. Revolve (169,97), pick the region, pick the centerline as axis, full 360, accept.

## Linear pattern (tested, partly)

Find via the toolbar or alt+c 'linear pattern'. Switch to Feature pattern, click the feature in the tree, set direction, distance and count (per the typing rules). Tick Centered to keep instances on the body. Verify the count after accepting.

## New document (tested)

1. Documents page: click 'Owned by me' in the left rail (the default view may be a school/company space).
2. Create (73,102), then Document... (78,127). Type the name, press Enter. Loads in 5-8 s with Part Studio 1 (100,789) and Assembly 1 (233,789) tabs.

## Part Studio layout (tested)

- Planes in tree: Top (64,197), Front (67,214), Right (66,232).
- Toolbar y=97: Sketch 102, Extrude 146, Revolve 169, Sweep 191, Loft 213, Fillet 274, Chamfer 308, Shell 388, Variables (x) ~855, Search tools 1160.
- Feature dialog accept = green check (145,120); red X (164,120) cancels.

## Shortcuts

- Tested: shift+7 iso, n normal-to-sketch, r corner rectangle, d dimension, alt+c tool search, ctrl+z undo. Plain e does NOT extrude.
- Shown in tool search, untested: shift+s sketch, shift+e extrude, shift+f fillet. From lessons: q construction, p planes, Esc exit tool; assembly y hide / shift+y show; drawings f fit, d dim, n note, ctrl+q update.

## Sketch on plane (tested)

1. Click the plane in the tree, click Sketch. Press n. (Top plane viewed via n can appear from below; use Front or the view cube if orientation matters.)
2. Sketch toolbar y=97: Line 135, Corner rect 169, Circle 204, Arc 238, Polygon 273, Spline 308, Point 343, Text 365, Fillet 447, Trim 482, Mirror 551, Dimension 646. Constraints x 672-967.
3. Accept with green check (145,120).

## Rectangle + dimensions (tested)

- r, click origin (snaps), click opposite corner. Do NOT type into the two live boxes (Tab entry over-defines). Press Esc/ctrl+z to drop them.
- Click Dimension button (646,97) (more reliable than d). Click edge, click empty space, type value, Enter. Tool stays active for the next edge. Units default inches; type '50 mm' for metric.
- Circles: click the circle's EDGE, not its interior.
- 'Cannot be created between these items' = wrong pick or tool not really active; click the toolbar button again. Red dimension + 'constraints not applicable' = undo.

## Extrude (tested)

Select the sketch, click Extrude. Mode tabs New/Add/Remove/Intersect at ~(41/71/106/150,157). Depth ~(120,231) with one region; it moves down (~115,247) when several regions are selected because the face list grows. Triple-click, type, Enter, then green check. - Multiple regions: click each region once; a second click deselects it. - The green check sometimes needs a second click; confirm the dialog closed before moving on.

## Variables (tested)

1. Variables button (~855,97). Type Name (e.g. thk) and Value (e.g. 10 mm). Green check (145,120). Tree shows #thk (converted to doc units).
2. Use it: in any value field type #thk, then CLICK the dropdown suggestion (~70,265 under the Depth field). Then accept.
3. Inline creation like '#dia=150 mm' in a sketch dimension FAILS (field goes red). Always create the variable first.

## Fillet / Chamfer (tested, partly)

Fillet (274,97), click edges, radius field ~(128,190) defaults 0.031 in; check it before accepting.

## Shell (tested)

Click the face to remove first, then Shell (388,97). Face auto-fills, thickness 0.1 in default at ~(140,193). Two-click shell.

## Circular pattern (tested)

Toolbar circular pattern, set type to Feature pattern, click the Features field, pick the feature in the tree, then CLICK THE AXIS FIELD FIRST and click a cylindrical face (bore). If the Features field is still active, that click adds the face's feature to the list instead (remove with its x). Set count, accept (145,120). The dialog grows over the tree, so pick tree items early.

## Mirror (tested)

Toolbar Mirror, set Feature mirror, pick the feature, click the Mirror plane field, pick the plane in the viewport (the tree is hidden). Front and Right look alike: hover to confirm the highlight before clicking.

## Chamfer (tested)

Chamfer (308,97), click the edge, Distance field ~(115,226): ctrl+a, type '1mm', read back, accept (145,120).

## Editing a feature (tested)

Double-click the feature in the tree to reopen its dialog. Linear pattern 'Could not create all instances' means the instances run off the body; tick Centered or shorten the distance.

## Assembly (tested: insert, fix, revolute, slider)

- Accept any pending Part Studio feature first, or Insert lists only some parts.
- Insert (126,97), Part Studios tab, click each part (~251, y=311/339/366); 'Inserted: N'. ACCEPT with green check (336,120). The red X (354,120) cancels ALL inserts.
- Fix: right-click the instance in the Instances tree (~77,197) > Fix (~113,360).
- Mate toolbar y=97: Fastened ~192 (unverified), Revolute 214 (shortcut m), Slider 237.

- Picking mate connectors: hover a face or edge until its connector dot shows, then click the DOT. Circular edges/faces give centered connectors (what Revolute needs). Bare corner vertices often don't highlight; use edge midpoints. The FIRST connector picked is the part that moves, the SECOND is where it goes (pick the fixed part second).
- Accept with the dialog check (145,120). A floating mini-toolbar near the parts has its own check, Solve and flip/reorient buttons.
- The motion axis is the connector's Z. A Slider on an edge midpoint of a top face slides vertically; for an in-plane slide pick connectors whose Z runs along the rail, use the reorient buttons, or use a Planar mate.
- Test motion by dragging along the allowed direction only.

## Drawings (partly tested)

- Create: '+' next to the bottom tabs (~37,789) > Create Drawing > template (default came up A3, first-angle) > pick the part. Place the base view, then projected views by moving the mouse up/right and clicking, plus an isometric.
- Thin parts can hide the projected-view ghost preview; reselect the parent view and try again.
- Drawing toolbar: Dimension ~(327,97) with a type dropdown at ~(342,97); Note 'A' ~(508,97) (unverified).
- KNOWN BLOCKER on the box browser (Oct 2026): every drawing dimension tool (Dimension, 2-point linear, diameter, d key) freezes the drawing page as soon as you click or hover geometry. Nothing gets saved; a page reload recovers it. Don't spend time retrying. Options: dimension the model with sketch dimensions and show them, add drawing dimensions from the user's own computer, or test whether Notes work first.
- PDF export (tested): right-click the Drawing tab (~240,789) > Export... (~225,757) > Format PDF, Download > Export (~758,509). The file lands in ~/Downloads within ~8 s.
- Use browser_take_screenshot after each click; Computer screenshots were stale here.

## Learning Center

- learn.onshape.com Sign In uses the existing Onshape session via sso.onshape.com.
- Paths: /learn/learning-path/onshape-fundamentals, /learn/learning-path/introduction-to-cad. Courses: /learn/course/<slug> (fundamentals-sketching, introduction-to-part-studios, introduction-to-onshape-assemblies, introduction-to-2d-drawings). Lessons: /learn/course/<course>/<module>/<lesson>?page=1.
- Progress comes mostly from video lessons and exercises; text lessons have Key Takeaways. Do exercises in a new personal doc named 'Learning - <lesson>'.

## Verification (bridge retro, tested-ish)

- View cube "Front" shows the Front plane; if the profile was sketched on Top, Front shows the thickness. Use "Top" (or sketch profiles on Front next time). Name which view you used in screenshots.
- Before typing a value, click inside the dialog field and confirm the cursor is there; stray keys go to viewport shortcuts (e.g. letters toggle tools). Press Esc to clear, re-click, retype.
- Measure tool is unreliable for the agent: if a selection fails twice, fall back to bbox via API + exported STL bounds (`/home/box/agent-data/workflows/engineering-cad-release-workflow/tools/stl_check.py`) and say that's the method.
- Laggy page: wait for the regen spinner to stop, take a fresh screenshot before every click; avoid rapid sequences.
- Zoom close-ups (scroll on the feature) of every foot, contact face, load point, node and re-entrant corner; look for nubs, dips, slivers. A small circle at the origin is the origin marker, not a hole — toggle origin visibility to confirm.

- Mass properties panel blank = no material (often after regeneration): re-assign and re-check.

## Measure (tested 2026-10-28)

- There is NO Measure button or tool search entry: measuring is automatic. Select entities, then read the bottom-right status bar.
- One entity shows its own value (e.g. a hole's cylindrical face shows "Diameter: 8 mm"). Select the cylindrical FACE, not the circular edge.
- Two entities: click the first, Shift-click the second, wait ~3 s. You get e.g. "Parallel axes dist / Min dist / Max dist".
- Do NOT click the status-bar icons; they open Custom tables / Appearance panels (this was the earlier "Measure opens other panels" failure).
- Values follow document units (inches by default); convert or switch the document to mm first.
- Part right-click menu shifts ~13 px between openings: screenshot before each click.
- Assign material > Custom tab: density is in the document's units (lb/in^3 in an inch doc; 1.27 g/cm^3 = 0.04588 lb/in^3). A blank mass in Mass properties means no material.
- "Onshape is not connected / Poor connection" freezes the UI and discards unsaved panel edits; reload, wait ~40 s, reopen the panel.
- View cube menu is the reliable way to Isometric; the `7` key may not register from automation.

## Verification tools (tested 2026-10-28, part 2)

- Units: hamburger menu > "Workspace units..." > Length Millimeter, Mass Gram, Density g/cm^3 > green check (wait ~1 s per dropdown).
- Mass properties: select the part, then the LAST icon in the bottom-right status bar ("Display mass and section properties"); not in the right-click menu. Verified 46,729.2 mm^3 x 1.27 = 59.346 g.
- Measure: two planar faces -> "Parallel dist"; two vertices -> Min/Max dist; one vertex -> XYZ; one face -> area. Hidden face: select first, rotate view, Shift-click second.
- Section view: select a plane in the Features tree > View menu > "Section view...".
- Profile view: right-click the sketch plane (e.g. Top) > "View normal to". Front/Right show a Top-sketched part edge-on.
- Zoom to selection: part right-click menu or View menu.
- Material assigned tell: part menu reads "Edit material..." instead of "Assign material...".
- Automation: send clicks one at a time; screenshots can lag one frame; screenshot context menus before clicking; accessibility fill() is more reliable than coordinate clicks in the Material panel.
- Not yet tested: Errors panel, Check interference, STL export name field (click into the field before typing).

# onshape-hole-feature-and-standard-content

**Summary (description):** Use when putting any fastener, tapped, counterbored, countersunk, PEM or bearing-adjacent hole into an Onshape part (including one hole through a stack of several parts), adding external threads, setting hole tolerances and callouts, or inserting, sizing, stacking and bulk-editing screws, nuts and washers from Onshape standard content in an assembly.

# Onshape Hole feature, External thread and standard content (deep reference)

> **Native features only, never FeatureScript.** Built-in Hole, External thread and Insert > Standard content only. No fastener/hole custom features, no Feature Studios. API-made Hole features failed before; add real Hole features in the **GUI** where callouts matter (onshape-api-rate-limits).

Labels: FACT = help pages hole.htm, external_thread.htm, standard_content.htm, hole_thread_callout.htm, tolerance_options.htm; INFERENCE; GUESS. Which size/fit to choose: mech-design-bolted-joints-and-fasteners.

## 1. Hole feature settings (FACT)

- **Placement**: sketch points, vertices, circle centres, implicit or explicit mate connectors. Selecting a whole sketch picks all its non-construction points, so make layout points construction if they aren't holes.
- **Merge scope**: which parts get cut. Several parts = one hole through the stack.
- **Style**: Simple, Counterbore, Countersink.
- **Type**: Drilled; **Clearance** (Fastener fit: Close, Free, Close (ASME), Normal (ASME), Loose (ASME)); **Tapped** (Straight, Straight pipe (metric), Tapered pipe; thread class); **PEM** (not with counterbore or countersink).
- **Start**: Start from part (default, smart start depth) / Start from sketch plane (also allows overlapping holes) / Start from selected plane or mate connector.
- **Termination**: Blind (depth includes the counterbore), Up to next, Up to entity (each with Offset from tip), Through all (+ Tap through all). Tap clearance is given in **pitches**. Tip angle 118°, 135°, Flat or Custom.
- **Tolerances** on diameter, depth, counterbore/countersink: Precision + Type, including Fit / Fit with tolerance / Fit (tolerance only) with ANSI or ISO hole and shaft classes.
- Threads are **cosmetic** (fast regeneration); don't model helical threads for real parts (INFERENCE).

## 2. Hole across multiple parts (FACT; this is the Quickstart "Hole across multiple parts" lesson)

With **two or more parts in Merge scope** and Type = Tapped with a Fastener fit other than None: the first parts get **clearance** holes and the **last** part gets the tapped hole. This replaces the old "Blind in last" option. Use it for: bracket screwed into a tapped block, lid into an enclosure boss, motor plate into a gearbox spacer. Order the selection so the tapped part is last (INFERENCE: check the preview colour).

## 3. Why always the Hole feature (FACT)

- Drawing **Hole/Thread callouts only work on Hole-feature holes**, not Extrude Remove circles; the callout pulls precision, tolerance, fit class and shaft class automatically.

- Standard content **Auto-size reads the Hole spec** (fine vs coarse thread, length). Non-Hole holes default to coarse and their length can't be detected.
- Overlapping holes drill at 0 depth: use Start from sketch plane, or split into two features.
- Hole tables on drawings come from Hole-feature holes (seen in the Explore Onshape "Jackhammer" drawing).

## 4. External thread (FACT)

- Cosmetic ANSI or ISO thread on a convex cylinder; end type Blind or Up to next; optional chamfer.
- Optional **undercut**: default diameter 0.015 in below the minor diameter, length 2x pitch.
- Shows on drawings with the Hole/Thread callout. Model the cylinder at the thread major diameter (INFERENCE).

## 5. Standard content (FACT)

- Libraries: ANSI and ISO bolts/screws, nuts, washers; DIN the same plus shaft keys; PEM.
- **Auto-size**: bolts round **down** (no interference), nuts round **up** from a bolt shaft.
- Fast placement:
- Pre-select a hole and click Insert; or pre-select a **planar face** to fasten into **all** its holes regardless of size.
- Press **a** to flip.
- Washers and nuts have two mates: **Insert closest** (Alt+Insert) / **Insert furthest** (Shift+Insert) to stack.
- A washer is auto-placed under a detected bolt head.
- Bulk change: right-click > **Select instances with same configuration**, then **Edit standard content instance** to change size and length together. Standard, category, class and component can't be changed in bulk.
- Part number and description can be set (company-wide on Professional).
- Pattern with **Replicate** (onshape-assemblies-and-mates), not one insert per hole.

## 6. Worked mini-example: electronics lid screwed into a printed enclosure (M3)

1. Enclosure Part Studio: bosses with layout points (non-construction) at the four lid-screw centres.
2. Printed PLA/PETG can't hold M3 threads (FACT Makelab: printed threads min M6 on FDM), so don't use a tapped multi-part hole here. Use Hole > Simple > Clearance M3, Fastener fit Free, Merge scope = **lid** only; and a Drilled hole in the boss sized for the heat-set insert from the insert drawing (mech-design-fits-and-tolerances §3).
3. Same parts in aluminium: one Hole, Tapped M3x0.5, Fastener fit Free, Merge scope = **lid, then boss**. The lid gets the clearance hole and the boss the tapped hole (FACT §2). Read the actual diameters in the dialog; set depth >= 2x D engagement (GUESS, 6 mm) plus tap clearance of 2 pitches.
4. Assembly: pre-select the lid top face > Insert ISO socket head M3 > Auto-size; check length; Replicate is not needed because the face selection filled all four holes.
5. Drawing: Hole/Thread callout on one hole + "4X" via the count type; tolerance comes from the feature.

## 7. Gotchas

- Selecting a layout sketch with many points makes holes at all of them: hide or make extras construction.
- Blind depth includes the counterbore depth: a 10 mm blind counterbored hole is not 10 mm of thread.
- Auto-size rounding down can leave a nut with too few threads: re-check protrusion (mech-design-bolted-joints-and-fasteners §5).
- Changing hole size later doesn't resize inserted fasteners automatically: bulk-edit them (INFERENCE).
- PEM type can't be combined with counterbore/countersink.

- GUI quirk seen on the box: opening Hole via tool search + Enter did nothing; clicking the Hole toolbar icon works.

## 8. Done when

- [ ] Every fastener hole is a Hole feature with type, fit and termination chosen on purpose.
- [ ] Multi-part holes use one feature with the tapped part last.
- [ ] Standard content placed by face or Replicate, sizes verified, callouts on the drawing.

# onshape-in-context-and-master-model

**Summary (description):** Use when a part's shape or holes must follow other parts in an assembly (side plates around bearings, a bracket that bolts to existing tube, a cover that wraps electronics, a desk crossbar between two legs), when choosing between Edit in context, Create Part Studio in context, multi-part Part Studio and a Derived master model, or when an in-context reference is stale, purple, broken or slow. Extends onshape-top-down-master-layout.

# Onshape in-context design and master model (from Learning Center + Help)

> **Native features only, never FeatureScript.** No custom features (Origin Cube, generators), no Feature Studios, no evalFeaturescript. GUI and built-in features only; API only lightly per onshape-api-rate-limits. Team documents are read-only: practise in a personal copy.

Labels: FACT = Onshape Help/Learning Center page, INFERENCE = reasoning from those facts, GUESS = rule of thumb to confirm.

## 1. Pick the method first (FACT, Learning Center "Managed In-Context" + help modeling_in_context)

| Situation | Method |
|---|---|
| Design intent clear up front, parts share one origin (gearbox plates + spacers) | Multi-part Part Studio |
| Several Part Studios must share driving numbers, envelopes, pivots | Master model: layout Part Studio + **Derived** (see onshape-top-down-master-layout) |
| A NEW part whose shape comes from parts already placed in the assembly | **Create Part Studio in context** |
| An EXISTING or linked part needs geometry from its neighbours, or geometry depends on position | **Edit in context** (scales well; FACT) |
| Part released / instance references a revision | Can't edit in context (FACT). Derive or make a new version instead |

INFERENCE: on a robot, use master model for the frame, pivots and bumper envelope; use in-context for the brackets and covers that just have to hug what is already there.

## 2. Master model data (FACT, "Master Model" course outline)

1. **Driving parameters**: variables (Variable Studio or a Variable feature) that size everything.
2. **Layout sketches**: points, lines and circles for pivots, axles and hole patterns.
3. **Space allocation**: simple solids reserving volume (battery, motor, wiring, hand clearance).

Space-allocation rules (FACT, course "Space allocation" video summary): - Keep the solids crude (blocks and cylinders), in their own Part Studio. - Derive from the **workspace** for auto-updates while designing; derive from a **version** for control. Cross-document links must use a version. - Avoid chains of Derived instances; derive whole Part Studios or sketches for performance. - Use configurations for quick variants (e.g. battery upright vs flat).

## 3. Edit in context: checklist (FACT unless marked)

- [ ] In the assembly, right-click the instance > **Edit in context** (or double-click it and choose a context). The assembly appears ghosted around the **primary instance**, which sets the placement.
- [ ] **Rename the context right away** (e.g. `Ctx Arm Stowed`); default names get confusing (INFERENCE).
- [ ] Reference ghost geometry with `Use` (u) or by selecting faces/edges. References show **purple**. An entity must be referenced, or the context will not persist after you close it.
- [ ] Multi-pose parts: in the context, tick **Follow position** and pick a **named position** (e.g. the arm at its hard stop). The context only goes stale when that position or the geometry changes.
- [ ] Too much ghosting? Right-click > **Hide context instance(s)**. Wrong variant? **Change configuration**.
- [ ] Making edits that have nothing to do with the assembly? Pick **None** in the Assembly contexts list.
- [ ] After the assembly changes: a blue dot means an update is available. **Updates are never automatic**: right-click the part > **Update context** (assembly) or the context's dot menu > Update context (Part Studio).
- [ ] If the wrong instance anchors the context: right-click > `<context>` > **Set as primary instance** (solid arrow = primary, dashed = secondary).

# 4. Create Part Studio in context: checklist (FACT, help create_part_studio_in_context)

- [ ] In the assembly, Insert dropdown > **Create Part Studio in context**.
- [ ] Choose the origin: an explicit **mate connector** on the mounting face, or the assembly origin. INFERENCE: a mate connector at the bolt-pattern centre gives a sensible part origin for drawings and CAM.
- [ ] Model the part, then finish with **Insert and go to assembly**. That makes the inserted instance the primary instance (solid arrow).
- [ ] Gotcha: **Go to assembly** followed by a separate insert gives a non-primary instance (dashed). Then use Set as primary instance.

# 5. Worked mini-example: intake side plate around COTS bearings (INFERENCE workflow)

Assembly has two 1.125 in OD flanged bearings on a 1/2 in hex roller, mated to a 2x1 tube. 1. Insert > Create Part Studio in context; origin = mate connector on the tube's outer face. 2. Rename the context `Ctx Intake Deployed`; Follow position = named position `Deployed`. 3. Sketch on the tube face: `Use` the bearing OD circles and the tube's bolt-hole edges. Offset an outline 0.375 in around them (wall GUESS; check stiffness in onshape-simulation-fea-sanity). 4. Extrude 0.25 in. Bearing bores: Hole feature or sketch circle with a fit (see mech-design-fits-and-tolerances). Bolt holes: Hole feature at the `Use`d centres (see onshape-hole-feature-and-standard-content). 5. Insert and go to assembly; fasten with standard content. 6. When the roller spacing changes later: Update context, then re-check the hole Fastener fit.

# 6. Gotchas

- The context is a **snapshot**. A colleague moving a bearing does not move your plate until you Update context (FACT).
- References to ghost **fillet edges** or imported mesh break easily; reference cylinder axes, planar faces and mate connectors (INFERENCE).
- Edit in context on a linked-document instance edits the **source** document; check edit rights first. Team docs are read-only here.
- Long chains (A references B, which references C) across documents get slow and fragile (FACT, course). Pull shared numbers into the master and Derive them instead.

- After a big re-layout, open each context once and confirm the purple references still land where intended.
- The Managed In-Context and Master Model lesson bodies need a Learning Center sign-in ("Purchase Required" text on lesson pages); only outlines and video summaries were readable.

## 7. Done when

- [ ] Every in-context part has a renamed context and a sensible primary instance.
- [ ] No unexplained blue "update available" dots before a version or release.
- [ ] Shared numbers live in the master (variables/layout), not in someone's context.
- [ ] A version exists before any cross-document Derive.

Related: onshape-top-down-master-layout (layout workflow), onshape-assemblies-and-mates (mates, named positions), onshape-release-and-versioning (versions before Derive).

# onshape-large-assembly-performance

**Summary (description):** Use when an Onshape robot or machine document gets slow — long Part Studio regeneration, slow mate solves, laggy graphics, too many tabs — or before building a large top assembly, to apply native fixes like version references, rigid/locked subassemblies, proxy parts, display states, a Lite configuration and document splitting, instead of the Simplify custom feature.

# Onshape large-assembly performance

> **Native features only.** Never use FeatureScript: no custom features (including **Simplify**, Origin Cube or any "lightweight" generator), Feature Studios, `/featurescript` or evalFeaturescript. Every speed-up here is a built-in tool used by hand.

## Thresholds to measure against

Performance panel thresholds (Onshape Help):

| Item | Threshold |
|---|---|
| Single feature regeneration | > 10 s warning |
| Part Studio total regeneration | > 30 s info, > 60 s warning |
| Features per Part Studio | > 500 warning |
| Assembly mate solve | > 5 s warning |
| Configured workspace inserts | regen > 15 s info, > 30 s warning |
| Tabs per document | > 500 warning (performance considerations page: keep < 100) |
| Part Studios + Assemblies per document | > 200 warning |
| Network latency | > 250 ms warning |

- Graphics culling (FACT): small instances are culled below 2 px on screen in assemblies under 5,000 bodies, and below 10 px above that. This is graphics only; the BOM is unchanged.
- Primitive count (observed practice, FRC 6329 on Chief Delphi): their public robot document, which loads quickly, sits around 4 million primitives in the Performance panel; robots above about 10 million slow down. Hiding complex-geometry parts or changing render appearance helps.
- Team working targets (INFERENCE): < 40 tabs per document; ≤ 10 parts or ≤ 250 features per Part Studio; top-assembly mate solve < 2 s.

## Workflow

1. **Measure first.**
2. Open the **Performance panel** for the document and read its warnings.
3. In each slow Part Studio, open the **Regeneration times** panel (clock icon in the Feature list) and sort by time.
4. In assemblies, note the mate solve time.
5. **Fix slow Part Studios:**
6. Move the slowest features (threads, textures, fillets on many edges, cosmetic details) to the **end** of the feature list.
7. In feature patterns and mirrors, turn **Reapply features** on only when needed.
8. Add a configuration (a "Lite" checkbox) that **suppresses** heavy cosmetic features. Design in Lite; turn it off for renders, drawings and exports.
9. **Pause regeneration** (roll back the bar) while editing configuration tables in big studios.

10. Split studios that exceed the targets, one mechanism or part family each.
11. **Derive from versions, not live workspaces.**
12. Version the master or layout and derive from that version (versions don't regenerate).
13. Don't daisy-chain derives: past three Part Studios in a chain, derive directly from a version.
14. When the design changes: branch → edit → version → update the derive.
15. **Fix slow assemblies:**
16. Insert parts and subassemblies **from versions** (version references load faster and don't auto-update).
17. **Fully constrain subassemblies (rigid)** so higher levels skip their mate solve. The rigid icon confirms it.
18. Where motion must stay, **lock** the subassembly instance or use **Lock/follow position to** a Named position.
19. Use the fewest mates that capture the motion (one Revolute rather than stacks of tangent, planar and parallel mates).
20. **Reduce mate limits** or suppress them until needed.
21. Group static clusters (Group) instead of mating each part.
22. **Replace Simplify with native proxies:** | Need | Native method | |---|---| | Light motor, gearbox or vendor STEP | Model a **proxy part**: a bounding-envelope cylinder or box with the real mounting holes, shaft and mate connectors. Keep the real part in a configuration or separate document. | | Hide internals for top-level work | **Display states** that hide fasteners, bearings and internals | | Fast top assembly | Insert **rigid** subassemblies from versions; **lock** moving ones | | Cosmetic detail toggle | "Lite" checkbox configuration suppressing threads, fillets, logos and textures | | Huge imported assemblies | Import as one document. Importing large assemblies as many separate documents "is not recommended". Replace the worst parts with proxies. | | Fasteners | Standard content with simplified/no thread display. Replicate instead of patterning many mates. |
23. **Split the robot across documents** (see `frc-robot-cad-organization`): one document per subsystem, the robot document linking versions of each, under 100 tabs (team target < 40).
24. **Graphics and viewing:**
25. Use **display states** and **named views**.
26. Don't force the highest tessellation on everything; set coarser tessellation on heavy parts.
27. Selecting an instance refines it first, so avoid select-all while zooming huge assemblies.
28. **Hardware and network:**
29. Discrete GPU and recent browser; check the hardware recommendations page.
30. Watch the **Poor connection** indicator; avoid VPNs; use the nearest region.
31. **Re-measure** with the Performance panel and regen times. Record before/after numbers in the version description.

## Proxy part recipe (native replacement for Simplify)

1. Insert the real vendor part (or open its STEP) in a scratch Part Studio. Measure the envelope, mounting bolt circle or pattern, shaft diameter and length, and the pilot boss.
2. In a "Proxies" Part Studio, sketch on Front: the envelope circle or rectangle plus the pilot boss. Extrude to the measured length.
3. Add the mounting holes with the **Hole** feature (tapped size as on the vendor part) and the shaft as a separate extrude.
4. Add explicit **mate connectors** at the face center (shaft axis) and at the mounting face, named `MC Shaft` and `MC Mount`.
5. Set the material or **mass override** to the vendor mass so CG and weight budgets stay correct.
6. Name the part `PROXY <vendor> <model>`, and set the part number to the vendor's number so the BOM stays right.
7. Version, then insert the proxy in assemblies. Swap to the real part with Replace instance only for renders or interference checks.
8. Check: overlay the proxy on the real part. Holes and shaft coincide within 0.1 mm; mass matches within 2 %.

## Do

- Measure before and after every change; fix the biggest number first.
- Keep subassemblies rigid unless motion is being studied.
- Put proxies and Lite configurations in place before the top assembly grows.
- Use version references everywhere above the subsystem level.

## Don't

- Don't install Simplify or any custom feature to "lighten" models.
- Don't import a vendor STEP with hundreds of bodies into the top assembly unprocessed.
- Don't leave Reapply features on in patterns by default.
- Don't put mate limits on everything "just in case".
- Don't derive through long chains of live Part Studios.
- Don't keep hundreds of tabs in one robot document.

## Common mistakes

- **Blaming the network** when one 40 s fillet feature is the cause. Read the regen panel first.
- **A rigid subassembly made flexible** for one motion study and never reset, so the top level slows down.
- **A Lite configuration that also suppresses mounting holes,** so mates break when toggled. Suppress only cosmetic features.
- **Proxies with the wrong mounting pattern.** Check holes and shaft against the vendor drawing.
- **Workspace references** in the top assembly that update and re-solve whenever anyone edits.
- **Culling mistaken for missing parts** in big assemblies. The BOM still has them.

## Checks before done

- [ ] No Performance panel warnings, or each remaining one is justified in a note.
- [ ] Every Part Studio < 30 s total regeneration; no single feature > 10 s.
- [ ] Top-assembly mate solve < 5 s (team target < 2 s).
- [ ] Subassemblies inserted from versions; rigid or locked unless intentionally moving.
- [ ] Heavy vendor parts proxied; a Lite configuration exists where useful.
- [ ] Tabs per document within target.
- [ ] No custom features anywhere (search the feature lists for non-native features).
- [ ] Before/after numbers recorded.
- [ ] Proxies match the real parts' mounting geometry and mass.

## Sources

- Performance considerations (regen panel, Reapply features, suppression configuration, derive from versions, three-studio rule, rigid subassemblies, Lock/follow, culling, < 100 tabs, mates and mate limits, latency): https://cad.onshape.com/help/Content/Home/performance_considerations.htm
- Performance panel thresholds: https://cad.onshape.com/help/Content/Home/performance_panel.htm
- Hardware and graphics: https://cad.onshape.com/help/Content/Home/hardware_and_graphics_performance_recommendations.htm ; display data and tessellation: https://cad.onshape.com/help/Content/Home/graphics_area_display_data.htm
- Display states: https://cad.onshape.com/help/Content/Assembly/display_states.htm ; Named positions: https://cad.onshape.com/help/Content/Assembly/named_positions.htm ; Group:

https://cad.onshape.com/help/Content/Assembly/group.htm ; Replace instance: https://cad.onshape.com/help/Content/Assembly/replace_instance.htm

- Importing (large assemblies as separate documents not recommended): https://cad.onshape.com/help/Content/Document/importing_files.htm
- Derived: https://cad.onshape.com/help/Content/PartStudio/derived.htm ; versions: https://cad.onshape.com/help/Content/Document/versions_and_history.htm
- Mass properties (mass override): https://cad.onshape.com/help/Content/View/mass_properties_tool.htm
- Tech Tip, rigid/lock subassemblies: https://www.onshape.com/en/resource-center/tech-tips/optimize-large-assembly-performance-rigid-lock-subassembly
- Tech Tip, performance panel: https://www.onshape.com/en/resource-center/tech-tips/tech-tip-how-to-use-the-performance-panel-in-onshape
- FRC 6329 primitive-count remark: https://www.chiefdelphi.com/t/team-6329-the-bucks-wrath-2024-robot-reveal-dial/456061
- Observed practice: the Onshape Explore Postman top assembly (few instances, Fastened subassemblies, configurations suppressing alternates) and Jackhammer instance folders.

# onshape-native-api-builds

> **Summary (description):** Use when reading Onshape data (features, mass properties, BOM, configurations, exports) through the REST API, or — only with explicit approval — scripting native-feature builds, assemblies or drawing edits in a document the user owns.

<!-- changelog (bridge retro, 2026-10-28): 1. Regeneration/sketch updates can mint a new part ID and drop name + material: re-check metadata after every update. 2. Clean outline rules (no overlapping polyline offsets at nodes; fillet with a Fillet feature). 3. Post-build automated checks: bbox/mass via API + stl_check.py/profile_check.py. -->

# Onshape native API builds

> **Native features only.** Never use FeatureScript: no `/featurescript` or `api-adv/fs` endpoint, no evalFeaturescript, no custom `featureType`, no Feature Studio elements, no "cube" tutorial feature. Every feature sent must be one a person could add from the toolbar.
>
> **Read-mostly.** Team-owned documents are read-only. Any write needs the user's explicit OK, must target a document they own (or a copy), and starts with a version.

## When to use

- Bulk reads: feature lists, mass, bounding boxes, BOMs, configuration tables, Where-used, exports.
- Repetitive native builds the GUI would make slow (many parts, patterns, parameter sweeps), once a write is approved.
- Drawing batch edits (see `onshape-professional-drawings` for the modify envelope and per-item check).
- Not for anything that needs FeatureScript. If no native feature can do it, do it by hand in the GUI or report that it can't be done.

## Basics

| Item | Value |
|---|---|
| Base URL | `https://cad.onshape.com/api/v{N}/` (Enterprise: `https://<company>.onshape.com/api`) |
| IDs | 24-char hex: `d` document, `w` workspace, `v` version (immutable), `m` microversion, `e` element |
| Path pattern | `/…/d/{did}/{wvm}/{wvmid}/e/{eid}` |
| Auth | API keys (Basic) or OAuth2 |
| Version writes | 405 — versions are immutable |
| Read-only key POST | 403 |

## Auth rules

1. **Use API keys from the vault or an OAuth app.**
2. Load them into environment variables at run time.
3. Never paste them into chat, files, logs or commits.
4. Don't create keys unless the user asks.
5. **In-page session calls** (same-origin `fetch` with relative URLs from a signed-in Onshape tab) are allowed only if the user explicitly authorized them for this task.
6. The script may pass the page's own XSRF cookie into the `X-XSRF-TOKEN` header on writes, but must never print, log, return, copy or store cookie, XSRF or token values.

7. Reload the tab after a run; the page may show "not connected" while the script runs.
8. If a call fails with 403, stop and report it. Don't route around it.
9. Prefer least-privilege scopes, read only, whenever no write is approved.

# Limits and errors

- **Rate limits** are per endpoint. A 429 response includes `Retry-After` and `X-Rate-Limit-Remaining`. Sleep at least `Retry-After`, then retry with backoff.
- **Annual call limits:** Free, Standard and EDU Student plans get 2,500 calls per user per year; Professional gets 5,000. A 402 means the annual limit is used up. 4xx/5xx responses don't count.
- Budget calls before scripting. Cache GETs and batch with `features/updates`.
- **Error codes:**

| Code | Meaning | Fix |
|---|---|---|
| 400 | Bad request | Check the body schema |
| 401 | Auth failed | Credentials |
| 403 | Scope or permission | Stop; don't escalate |
| 404 | Wrong ID | Check the ID |
| 405 | Write to a version | Use a workspace |
| 409 | Conflict or microversion skew | Re-read and retry |
| 415 | Bad JSON or media type | Set Content-Type |

# Workflow

1. **Spec table first.** Turn every dimension into rows of (feature, parameter, value, units) before any call. Get the approval wording for writes.
2. **Resolve IDs** from the URL or `GET /documents/{did}`. Then `GET /documents/d/{did}/w/{wid}/elements` and record the element list as a baseline.
3. **Read before writing:**
4. `GET /partstudios/d/{did}/{wvm}/{wvmid}/e/{eid}/features` returns the features, sketch entities and constraints, and the rollback index.
5. Copy the parameter schema of an existing **native** feature of the same type. Never invent a `featureType`.
6. **Version the undo point:** `POST /documents/d/{did}/versions` with a clear name, e.g. `pre-API build <topic>`.
7. **Idempotency.** Match existing features by name. A rerun must update, not duplicate. Give every feature a unique, meaningful name.
8. **Writes, one feature at a time:**
9. `POST …/features` adds a feature (BTFeatureDefinitionCall-1406 wrapping BTMFeature-134).
10. `POST …/features/featureid/{fid}` edits one in place.
11. `POST …/features/updates` changes several.
12. Always send `sourceMicroversion` and `rejectMicroversionSkew: true` so a concurrent edit isn't overwritten.
13. **Check every response:**
14. Read `featureStatus` (OK / WARNING / ERROR); a 200 doesn't mean it regenerated.
15. Check the body count or mass after each feature. Stop on the first ERROR.
16. **Native feature menu:**
17. Sketch (BTMSketch-151), fully constrained
18. Extrude, Revolve, Sweep
19. Hole (standard and tap callouts, so drawings link to them)
20. Fillet, Chamfer, Shell
21. Linear/Circular pattern, Mirror
22. Sheet metal model, Flange, Finish
23. Derive, Variable
24. Part properties (name, number, material) through the metadata endpoints.

25. **Configurations:**
26. Encode values through the `configurationencodings` endpoints. Use the `encodedId` in request bodies and the `queryParam` in URLs, e.g. `configuration=List_xxx%3D_500_mm`.
27. Pass the configuration on **every** call, or you edit or read the default.
28. **Assemblies:**
    - Insert instances from versions (or same-document Part Studios). Placement transforms are 4×4 row-major matrices in metres.
    - Native mates (Fastened, Revolute, Slider) between named mate connectors.
    - One fixed instance. Fixing an existing instance has no API call; do it in the GUI.
29. **Verify:**
    - `GET …/massproperties` (Part Studio or assembly) against the hand-calculated mass (±2 %).
    - Bounding box against the envelope.
    - Hole count against the spec.
    - Compare the element list with the baseline: only assigned elements changed.
30. **Export:**
    - POST a translation, then poll `requestState` (ACTIVE → DONE or FAILED) every few seconds with backoff, or use a webhook.
    - Download the result from the resulting external data or blob element.
31. **Version again** (`<topic> built`) and report the version ID and any WARNING features.

## Testing before a live run

- Build the script against a mock Onshape (a fake fetch that records calls and returns realistic shapes) with scenarios: fresh run, idempotent rerun (no writes), server error 500 (stops with diagnostics), wrong document (stops before any call), and partial failures.
- In every scenario assert: URLs are relative, credentials are included, the XSRF header is sent only on writes and never appears in results or logs, and no FeatureScript or Feature Studio endpoint is called.
- Run `node --check` on the bundle.

## Read recipes (no approval needed for documents you can view)

- **Weight check:** massproperties per Part Studio, summed against the budget.
- **Audit for custom features:** GET features and flag any feature whose type isn't a built-in one, plus any Feature Studio element in the element list. Report them; never delete without approval.
- **BOM:** assembly BOM endpoint into a CSV of part numbers and quantities (assembly endpoints per api-adv/assemblies).
- **Configured variants:** read the configuration definition, then loop over the encodings for mass and bounding box.

## Do

- Keep calls sequential within a Part Studio (the feature list is ordered); different elements can run in parallel.
- Send values as expressions with units (`"12 mm"`, `"90 deg"`), never bare numbers in metres.
- Edit sketches in place so downstream references survive.
- Log the call count against the annual budget.

## Don't

- Don't call `api-adv/fs` or any FeatureScript eval, even read-only.
- Don't write to team-owned or public documents; copy first, with approval.
- Don't delete features without versioning and listing their dependents (deletes cascade).

- Don't retry a 403 by switching credentials or methods.
- Don't hammer the API on 429.

## Common mistakes

- **Treating a 200 as success** without reading `featureStatus`.
- **Recreating a sketch,** so its IDs change and downstream features break.
- **Forgetting the configuration parameter,** so the default configuration gets edited.
- **Ignoring the rollback index:** features insert at the end.
- **Bare floats** interpreted as metres.
- **Writing to a `v/` path** and getting 405.
- **Burning the 2,500-call EDU budget** on polling loops.

## After any update or regeneration (learned the hard way)

- Updating sketches/features can re-create the part with a **new part ID**; part name, material and appearance attached to the old ID are lost and mass properties go blank. After every update: list parts, re-check name + material + mass; re-assign material (prefer doing it as the last step, then re-verify).
- Build profiles as clean unioned regions; overlapping bar ends and polyline offsets/fillets at nodes create nubs below contact faces and V-dips on edges. Check the planned profile with `/home/box/agent-data/workflows/engineering-cad-release-workflow/tools/profile_check.py` before the build.
- After build: bbox + mass properties via API vs plan, export STL and run `/home/box/agent-data/workflows/engineering-cad-release-workflow/tools/stl_check.py` (watertight, volume, exact length, nothing below contact planes). Then the GUI check in `engineering-cad-release-workflow` Gate 6.

## Checks before done

- [ ] Approval recorded for every write; the target document is user-owned or a copy.
- [ ] No FeatureScript endpoint used; no custom featureType; no Feature Studio created.
- [ ] Versions exist before and after; the version IDs are reported.
- [ ] Every feature is named with status OK, or WARNINGs are explained.
- [ ] Mass, bounding box and hole count are verified.
- [ ] The element list matches the baseline apart from the assigned elements.
- [ ] No credential values appear in output, files or logs.

## Sources

- API intro (IDs, base URL, auth): https://onshape-public.github.io/docs/api-intro/
- Limits (rate, annual, 402/429): https://onshape-public.github.io/docs/auth/limits/
- Errors: https://onshape-public.github.io/docs/api-adv/errors/
- Feature access (features GET/POST, sourceMicroversion, rejectMicroversionSkew): https://onshape-public.github.io/docs/api-adv/featureaccess/
- Part Studios: https://onshape-public.github.io/docs/api-adv/partstudios/
- Assemblies: https://onshape-public.github.io/docs/api-adv/assemblies/
- Configurations: https://onshape-public.github.io/docs/api-adv/configs/
- Translation: https://onshape-public.github.io/docs/api-adv/translation/
- The FS endpoint is read only to know what to avoid: https://onshape-public.github.io/docs/api-adv/fs/
- The ±2 % mass tolerance and the poll interval are INFERENCE (team practice).

# onshape-part-modeling-workflow

> **Summary (description):** Use when modeling any part or multi-part Part Studio in Onshape (plates, brackets, shafts, housings, 3D-printed parts) and you need a clean, fast, drawing-ready feature tree built from native features only.

# Onshape part modeling workflow

> **Native features only.** Never use FeatureScript in any form: - no custom features, even public ones (Origin Cube, Simplify, laser-joint generators, belt/chain generators, "parametric part number" features, tube converters, lighteners); - no Feature Studios; - no FeatureScript API endpoint (`/featurescript`, evalFeaturescript).
>
> Build everything with the built-in toolbar features, the way a person working by hand would.

## When to use

- Creating a new part, or a multi-part Part Studio whose parts share design intent (gearbox plates, a frame, a housing and its cover).
- Cleaning up or reviewing an existing Part Studio before it is linked, drawn or released.
- Not for:
- layout and master sketches (see `onshape-top-down-master-layout`);
- sheet metal (see `onshape-sheet-metal`);
- freeform shapes (see `onshape-surfacing`).

## Workflow (do these in order)

1. **Decide the Part Studio scope.**
2. One Part Studio per *group of parts that change together*. Target ≤ 10 parts or ≤ 250 features.
3. Never put a whole robot or machine in one Part Studio.
4. Every identical part that appears more than once must be **one** part, patterned later in the assembly.
5. **Name the tab with the project scheme before you model** (see `frc-robot-cad-organization`).
6. Observed in the Onshape Explore "Postman" robot: the tab name carries the number, e.g. `(PRJ-PS100) Drivetrain Frame` for a Part Studio, with its matching `(PRJ-A100) Drivetrain` Assembly.
7. **Build the top of the feature tree, in this order:**
8. A folder `Variables`: every driving value as a Variable feature (`#plate_t = 0.188 in`).
9. Reference planes, named for their function (`Wheel Axis Plane`, `Frame Top Plane`).
10. A `Reference` or `Master Sketches – Derived` folder: Derived layout sketches from the master layout, or a local layout sketch.
11. A local reference sketch: fully defined, and driven by the derived geometry.
12. **Model one part per feature folder.**
13. Name each folder after the part (`Right Arm Joint Mount`, `Side Plate Features`).
14. In each folder: base Extrude or Revolve → secondary cuts → **Hole** features → patterns → part-specific fillets and chamfers.
15. Observed: Postman's arm Part Studio has one folder per part (e.g. `Smallest Rear Link (25)`). The drivetrain groups features by plate (`Side Plate Features`, `Front Plate Features`, `Top Plate Holes/Cuts`).
16. **Build one side, then Mirror.**
17. Model the right-side parts and Mirror them about a named center plane. Rename the mirrored parts `… L` / `… R`.
18. Observed in Postman: `Right Inner Mirror` / `Left Mirror`, and a mirrored wheel (left to right).

19. **Make every fastener hole with the Hole feature.**
20. Types: Simple, Counterbore, Countersink, Tapped, using the standard sizes.
21. Rename the Hole feature to its spec. Observed in the Onshape Jackhammer model: `Ø 9 mm ↧ 20 mm`, `M8x1.25 ↧ 35 mm`.
22. Hole features feed: drawing hole callouts and the hole table; Standard content Auto-size in assemblies; and the Part Studio hole table.
23. **Make joints with native features.**
24. For laser-cut tab/slot joints, draw the tab or slot profile once in a sketch, driven by variables (`#tab_len`, `#slot_w`, `#edge_off`).
25. Extrude-add the tabs and Extrude-remove the slots, then Linear or Curve pattern them along the edge.
26. Postman used a custom "Laser Joint" feature for this; **we don't.**
27. **Add details last.** Pockets/lightening cuts, cosmetic fillets and chamfers, and modeled threads (External thread) go at the end of the tree. They are slow and they break most easily.
28. **Fill in part properties.** For each part: a descriptive name, Part number, Description, Material, and Appearance (a color by subsystem helps reviews).
29. Configured families: set configured part numbers and materials (see `onshape-variables-and-configurations`).
30. Part numbers are typed in the Properties dialog or set as configured properties. Never use a custom part-number feature (Postman's "Parametric Part Numbers" is a FeatureScript feature).
31. **Check, then version.**
    - Run the checks below. Then create a version (e.g. `D1 <part> detail`) before anyone links, derives, draws or releases from it.

## Feature choices (native)

| Need | Use | Avoid |
|---|---|---|
| Plate, block | Sketch + Extrude (Shift+e) | Many tiny extrudes stacked |
| Shaft, spacer, pulley body | Revolve (Shift+w) of a half-profile | Extruded circles plus cut grooves |
| Fastener hole, tapped hole, counterbore | Hole feature | Extrude-remove a circle |
| Hole pattern | Hole on sketch points (one feature, many holes), or Linear/Circular pattern of the Hole feature | Repeated single holes |
| Identical parts | One part, patterned in the assembly (Linear/Circular pattern, Replicate) | Part-pattern inside the Part Studio (each copy becomes a separate BOM line) |
| Left/right parts | Mirror (part or features) about a named plane | Redrawing the other side |
| Shells, covers | Shell, then Hole/Extrude | Hollowing with many cuts |
| Ribs | Rib feature from a sketch line | Thin extrudes |
| Tubing, extrusions, weldments | Native Frame with a Frame profile from the Onshape library or a tagged custom profile, plus Frame trim and a Cut list | "Tube converter" custom features |
| Combining or splitting bodies | Boolean (Union/Subtract/Intersect), Split, Delete part | Overlapping parts left un-booleaned |
| One BOM line for several bodies | Composite part (closed) | Composite of hardware (loses the counts) |
| Threads | Tapped Hole and External thread | Helix sweeps (slow, and not linked to callouts) |
| Engraved text or logo | Sketch text + Extrude remove (shallow) | Wrap and decals on structural parts |

## Observed COTS modeling pattern (Jackhammer "Ball Bearing" Part Studio)

- One fully defined profile sketch, then separate Revolves for the inner race, outer race, ball and cage, then a Circular pattern of the balls.
- The result becomes **one Composite part**, so the bearing is a single BOM line.

- A **configuration list named by vendor designation** (6001, 6201, 6203, 6205, 6302) drives the variable-driven sketch dimensions.
- Model a COTS item once, configured; don't make one Part Studio per size.
- Wires or non-parametric imports go in their own Part Studio as a composite (Jackhammer "Internal Wires", 0 features), kept separate from parametric parts.

## Do

- Drive every important dimension from a variable or the layout sketch. Type expressions with units: `#plate_t * 2 + 0.5 in`.
- Name every feature that matters (Shift+n) and every part. Unnamed `Extrude 37` and `Part 12` are review failures.
- Put features into named folders so the tree reads like a build plan.
- Sketch on named planes or on stable faces of base features. Never sketch on fillet faces or faces created by late detail cuts.
- Use the rollback bar to insert features mid-tree. Put it back at the end afterwards.
- Fix errors from the **top** down: one failing early feature causes many later errors.
- Use the feature-list filters: `:errors`, `:warnings`, `:suppressed`, `:variable`, `:part`, `:name`, `:folder`, `:hidden`.
- Assign a material to every part. Parts without a material have zero mass, so weight and CG totals are wrong.
- Use Measure (`[`) and Section view (Shift+x) to check clearances while modeling.

## Don't

- Don't model all parts of a machine in one Part Studio. Onshape states this is bad for performance.
- Don't pattern identical parts inside a Part Studio. Each copy becomes a separate, unique BOM line.
- Don't fillet or chamfer sheet-metal parts with solid tools. Use Sheet metal corner break.
- Don't reference other parts' detail edges, fillets or pattern instances.
- Don't leave sketches under-defined (blue geometry).
- Don't leave orphan sketches, suppressed experiments or dead features. Delete them, or move experiments to a branch.
- Don't use FeatureScript custom features, even if a tutorial or forum post recommends one.

## Common mistakes

- **Holes made by Extrude-remove.** They give no callout, no Auto-size and no hole table. Replace each with a Hole feature at the same sketch point.
- **Part patterns inside the Part Studio.** They inflate the BOM and the instance count. Delete the pattern and pattern in the assembly.
- **Fillet before a pattern or mirror.** The feature fails or slows down. Move cosmetic fillets to the end.
- **Mirrored part left with the same name**, e.g. two parts called "Side Plate". Rename with an L/R suffix and give each its own part number.
- **A variable placed below its first use.** The feature errors. Variables are features, and order matters.
- **Material missing**, so the mass report undercounts.
- **Unit mixing in expressions** (`3 + 2.5 in`) or implicit multiplication (`(2)(3)`).
- **Long, chained dependencies**: one sketch driving the next, and so on. Prefer a flat tree where each part references the layout directly.

## Checks before done

- [ ] Performance panel: no feature over 10 s; total regeneration under 30 s (Onshape warns at 60 s); under 500 features.

- [ ] `:errors` and `:warnings` filters show nothing. Rollback bar at the end.
- [ ] Every sketch is fully defined (black).
- [ ] Every fastener hole is a Hole feature named by its spec.
- [ ] Every part has a name, part number, description, material and appearance. Mass properties show no missing-material parts.
- [ ] No in-Part-Studio patterns of identical parts. Mirrored parts are renamed L/R.
- [ ] No custom features and no Feature Studio tabs in the document.
- [ ] The document is versioned with a meaningful name before linking or drawing.

## Sources


- Part Studio basics, feature-list filters, "don't model the whole assembly in one Part Studio": https://cad.onshape.com/help/Content/PartStudio/part_studios.htm
- Hole feature: https://cad.onshape.com/help/Content/PartStudio/hole.htm ; Part Studio hole table: https://cad.onshape.com/help/Content/PartStudio/hole_table.htm
- Part patterns make unique parts (not recommended): https://cad.onshape.com/help/Content/PartStudio/circular_pattern.htm , https://cad.onshape.com/help/Content/PartStudio/curve_pattern.htm
- Mirror: https://cad.onshape.com/help/Content/PartStudio/mirror.htm ; Boolean: https://cad.onshape.com/help/Content/PartStudio/boolean.htm ; Composite part: https://cad.onshape.com/help/Content/PartStudio/composite_part.htm
- Frame, frame library, tag, cut list: https://cad.onshape.com/help/Content/PartStudio/frame.htm , https://cad.onshape.com/help/Content/Library/onshape_frame_profiles_library.htm , https://cad.onshape.com/help/Content/PartStudio/tag.htm , https://cad.onshape.com/help/Content/PartStudio/cut_list.htm
- Variables: https://cad.onshape.com/help/Content/PartStudio/variable.htm ; numeric fields: https://cad.onshape.com/help/Content/Home/numeric_fields.htm
- Repair top-down: https://cad.onshape.com/help/Content/Document/repairing.htm
- Mass/material: https://cad.onshape.com/help/Content/View/mass_properties_tool.htm , https://cad.onshape.com/help/Content/PartStudio/customizing_part_materials.htm
- Sheet-metal fillet/chamfer → corner break: https://cad.onshape.com/help/Content/PartStudio/fillet.htm , https://cad.onshape.com/help/Content/PartStudio/chamfer.htm
- Performance panel thresholds: https://cad.onshape.com/help/Content/Home/performance_panel.htm
- Learning Center curriculum outline (Onshape Fundamentals: CAD): https://learn.onshape.com/learning-paths/onshape-fundamentals-cad
- Observed practice: Onshape Explore sample documents "Postman" (FRC-style robot) and "Jackhammer", reached from the Onshape "Explore Onshape" page and studied read-only.

# onshape-professional-drawings

> **Summary (description):** Use whenever a part or assembly needs a drawing a shop could quote and make from — detail, assembly, sheet-metal or wiring sheets in Onshape with templates, views, toleranced dimensions, hole callouts and tables, GD&T, notes, BOM and balloons — using the GUI for everything and the Drawings REST API only for repeatable, approved batch work.

# Onshape professional drawings

> **Native features only.** Never use FeatureScript: no custom features, Feature Studios, `/featurescript` or evalFeaturescript. Drawings come from native Hole features, properties, exploded views and the built-in drawing tools. API use follows `onshape-native-api-builds`: credentials from the vault only, explicit approval for every write, and no writes to documents you don't own.

## When to use

- A part is about to be machined, laser/waterjet cut, bent, printed or bought as a modification.
- An assembly needs a build sheet: exploded view, BOM, balloons, torque notes.
- A checker or mentor review of an existing drawing.

## Workflow

1. **Prepare the model** (this is what makes drawings fast and self-updating):
2. Fill Part number, Description, Material, Revision (and Finish, if your template has it) in each part's Properties. The title block reads them.
3. Every hole is a **Hole** feature (simple, counterbore, countersink, tapped), so callouts link. Name holes by spec, as the Jackhammer sample does: `Ø 9 mm ↧ 20 mm`, `M8x1.25 ↧ 35 mm`.
4. Sheet metal is a native Sheet metal model with a flat pattern.
5. Assemblies have an exploded view, named positions and a complete BOM.
6. **Create a version** (`for drawings rev A`).
7. **Plan on paper first.** For each part list:
8. sheet size, scale and views;
9. datums (A = primary seating face; B and C = locating features);
10. each dimension with its tolerance **and the function that sets it**;
11. every hole callout; material, finish and notes.
12. **Create the drawing:**
13. Right-click the part (or the tab) → Create drawing of…
14. Template: ANSI (third-angle) for US work, ISO (first-angle) for ISO work, or the team template. Keep templates in one dedicated document.
15. Size: the smallest sheet that leaves about 30 % white space at a standard scale. Insert views: one, four or suggested.
16. **Drawing properties** (gear icon):
17. Units and precision; dimension text ≥ 3–3.5 mm; projection with the symbol shown; hidden lines off; tangent edges off; BOM style.
18. **Views:**
19. The front view shows the most shape with the fewest hidden lines. Add projected views (p) only as needed, usually 2–3.
20. Add a **section** through every counterbore, blind hole and pocket (`SECTION A-A`).

21. Add a **detail** for small features (`DETAIL B 2:1`), an **auxiliary** view for inclined faces, a **break** view for long uniform parts (the dimension still reads true length), and a **broken-out** section for one local hidden feature.
22. Add an **iso** for orientation only; never dimension it.
23. Add center marks on every hole and centerlines on every axis.
24. **Dimension from datums:**
25. Each feature once, in its clearest view.
26. No double dimensioning; leave one chain dimension open or make it reference `( )`.
27. No dimensions to hidden lines; no dimensions on the iso.
28. Hole patterns: ordinate or baseline from the datum, or basic dimensions plus position. No chains.
29. Spacing: about 10 mm from the outline and 6 mm between rows; shorter dimensions inside longer ones.
30. **Holes:**
31. Hole callout tool on the hole edge, with count type set so it reads `4×`; Add leader for extra same-size holes.
32. **Hole table** (Insert → Hole table, pick an origin and axes) for hole-heavy plates: TAG / X / Y / SIZE / QTY, as on the Jackhammer drawing.
33. Clearance per ISO 273 medium: M8 → Ø9, M6 → Ø6.6, M5 → Ø5.5, M4 → Ø4.5.
34. Tapped: `M8×1.25 - 6H ↧ 20` plus the tap drill on blind holes.
35. **Tolerances by function:** | Function | Typical spec | |---|---| | Non-functional outline | ISO 2768-c or ±0.5 mm | | General machined | ISO 2768-m (±0.1 at 6–30, ±0.2 at 30–120, ±0.3 at 120–400 mm) | | Floating-fastener hole pattern | Position Ø(H − F) at MMC (Ø9 holes, M8 → Ø1.0 Ⓜ) | | Fixed fastener (clearance to tapped) | Position Ø(H − F)/2 each part (→ Ø0.5 Ⓜ) | | Bearing or pin seats | ISO 286 fit: H7/g6 sliding, H7/k6 locating, H7/p6 press | | Mounting-face flatness | 0.05–0.1 over the face | | Laser/waterjet sheet | ±0.1 in one flat face; ±0.3 across a bend; bend angle ±0.5–1° | | FDM print | ±0.2–0.3; holes print 0.1–0.2 undersize; "drill/ream to size" on critical holes |
36. Prove every tight tolerance with `engineering-math-proofs` (fits, fastener position, stack-up).
37. **GD&T:**
38. Datum symbols on the real mounting and locating features (A 3 points, B 2, C 1).
39. Feature control frames: position for hole patterns, flatness for seating faces, perpendicularity for bores to a face, profile for complex outlines.
40. Locating dimensions are Basic (boxed).
41. Example from the Jackhammer drawing: `⌖ Ø.050 | A | B ⌖`, toleranced diameters `Ø69.01/68.99`, chamfer `1.01 × 45°`.
42. **Surface finish:** a general symbol near the title block plus symbols on functional faces (Ra 1.6 seating, 0.8 bearing bores, 3.2 general machined).
43. **Notes block (n), numbered**, above or beside the title block:
    1. INTERPRET PER ASME Y14.5-2018 (or ISO 8015)
    2. MATERIAL: AL 6061-T6 (with spec)
    3. FINISH: …
    4. GENERAL TOLERANCES ISO 2768-mK UNLESS OTHERWISE SPECIFIED
    5. BREAK ALL SHARP EDGES 0.3 MAX, REMOVE BURRS
    6. Part-specific notes: thread class, flatness, print orientation, bend data
44. **Title block** (as on the Jackhammer ANSI sheet): part number (DWG NO), title noun-first, revision, sheet n of m, scale (e.g. 1:2; iso views 1:4 labelled), size, projection symbol, units, material, finish, weight from mass properties, general tolerance, drawn/checked/approved, a zone grid (A–D / 1–4), and a revision table. Link fields to properties (right-click → Formats (unlock) → Edit note → Insert drawing property).
45. **Assembly sheets:**
    - Exploded view (choose the explosion in the view dialog) plus an assembled iso.
    - **BOM table** (Flattened or Structured) in a corner: Item, Qty, Part number, Description, Material.
    - **Balloon** tool on a visible edge of each part. Balloons are linked to the BOM item numbers. Align them in rows or columns.
    - Overall and stroke dimensions, a torque table, and assembly-order notes.

46. **Sheet-metal sheets:** folded views plus a flat pattern view (Insert dialog) with UP/DOWN bend notes, radius and angle, the overall flat size (REF), and the K-factor or bend-deduction basis. See `onshape-sheet-metal`.
47. **Wiring sheets:** a schematic from drawing-sketch lines and notes, a cable-list table (ID, from, to, gauge, connector, length), and a routing view.
48. **Update (Ctrl+q), then fix dangling (red/orange) annotations.** Version the document to freeze the drawing.
49. **Export:** right-click the tab → Export: PDF (selectable text, all sheets); DXF/DWG for laser shops (splines as polylines). Read the PDF back before sending.

## API vs GUI (Drawings API, formatVersion 2021-01-01)

| Task | API | GUI |
|---|---|---|
| Create a drawing (template, size, standard, projection, units, initial views, explosion, display state, flat) | Yes: `POST /drawings/d/{did}/w/{wid}/create` | Yes |
| Title-block values | Indirect: set part/document properties via metadata | Yes |
| Title-block layout or template | No | Yes |
| Base (TopLevel), Projected, Section, Auxiliary views | Yes: `onshapeCreateViews` / `onshapeEditViews` | Yes |
| Detail, broken-out, break, crop, aligned section views | **No** | Yes |
| Isometric | Yes (TopLevel with iso orientation) | Yes |
| Linear, angular, radial, diameter, chamfer, ordinate dimensions | Yes (`Onshape::Dimension::*`) | Yes |
| Baseline dimensions | **No** (use ordinate) | Yes |
| Tolerances (symmetric, deviation/limits, basic) | Yes; "Fit" via API is undocumented (UNVERIFIED) | Yes, including fits |
| Hole/thread callout (native Hole features only) | Yes: `Onshape::HoleCallout` | Yes |
| Hole table | **No** | Yes |
| GD&T frame, datum, surface finish, inspection symbol | Yes | Yes |
| Weld symbol | **No** | Yes |
| Centermarks, centerlines, sheet sketch lines/arcs/circles | Yes | Yes |
| Notes; circle "callouts" | Yes (callouts are **not** BOM-linked) | Yes |
| BOM table | Yes: `Onshape::Table::BomTable` | Yes |
| BOM-linked balloons | **No** | Yes (Balloon tool) |
| General table (wire list, torque table, manual revisions) | Yes: `Onshape::Table::GeneralTable` | Yes |
| Linked revision table | **No** (needs release management) | Yes |
| Delete annotations or views | Yes: `onshapeDeleteEntities` | Yes |
| Read the drawing | Yes: DRAWING_JSON translation, `…/views`, `…/jsongeometry` | – |
| Export PDF/DXF/DWG | Yes: translations, with `formatName` from `GET …/translationformats` | Yes |
| - **Summary:** the API covers about 80 % of a standard sheet. A good split: create the drawing, views and dimensions by API from the JSON export; finish detail/break views, hole tables, balloons, welds and the template by hand; export by API. | | |

## API batch procedure (only with explicit write approval, on a document you own)

1. **Version first.** Record every `logicalId` your run creates, so a failed batch can be cleaned up.

2. Read the drawing: POST a translation with `formatName: DRAWING_JSON`, poll `GET /translations/{tid}` (back off), and download it. Copy the `uniqueId`, `deterministicId` and `viewId` of the edges and points you need. Model coordinates are in metres; free text positions are in sheet units.
3. Modify: `POST /drawings/d/{did}/w/{wid}/e/{eid}/modify` with `{"description": "...", "jsonRequests": [{"messageName": "onshapeCreateAnnotations", "formatVersion": "2021-01-01", "annotations": [...]}]}`.
4. messageName is one of onshapeCreateAnnotations, onshapeEditAnnotations, onshapeCreateViews, onshapeEditViews, onshapeDeleteEntities.
5. Use `sheetDetails` to target a non-active sheet.
6. **Per-item batch-result check:** poll `GET /drawings/modify/status/{mrid}` and read *every* item's `status` (OK / Partial_success / Failed) with its `logicalId` or `errorDescription`.
7. **A batch returns HTTP 200 if any item succeeded**, so 200 alone means nothing.
8. Treat a batch as passing only when every item is OK. Retry or fix the Failed items individually, and log the Partial_success ones for a GUI fix.
9. Verify: export DRAWING_JSON again; every new logicalId exists; no `isDangling: true`; every view's `errorState` is empty. GD&T prefixes and suffixes aren't exported, so check those by eye on the PDF.
10. Delete only logicalIds your own run created, and only after a version exists.

## Observed assembly-drawing convention (Jackhammer "Drawing 1")

- **Sheet:** A2 ISO, scale 1:4, first-angle projection symbol, zone grid 1–8 / A–F, 1 of 1.
- **One exploded isometric view** of the whole assembly, with explode lines, in color appearance. **No dimensions;** those live on the part drawings.
- **Balloons:** item number plus a quantity prefix (`10x 24`), aligned in columns around the view.
- **Auto BOM table** at the left, columns ITEM, NAME, PART NUMBER, QUANTITY, UNIT OF MEASURE, MASS, MATERIAL, CATEGORY, sorted by item.
- **Revision table** at top right: Revision, Revision description, Date approved, Approver.
- **Title block** fed from document properties: TITLE, DWG NO, size, scale, sheet, material, mass, and DRAWN / CHECKED / APPROVED rows.
- **Notes:** units, angular ±1°, surface finish, DO NOT SCALE DRAWING, BREAK ALL SHARP EDGES AND REMOVE BURRS.
- **Where drawings live:** in the same subsystem folder as their Part Studios and assembly (Housings folder: assembly, Part Studios, "Gear Cover Drawing 1").

## Do

- Fill properties on the part, not as typed title-block text.
- Use Hole features and linked callouts; use hole tables for dense plates.
- Tolerance by function; keep everything else under the general tolerance.
- Keep all drawing work in one workspace or branch: drawings can't be merged, only Kept or Replaced.
- Record the document version used on the drawing (revision or note).

## Don't

- Don't type hole text by hand; it goes stale when the hole changes.
- Don't use API circle callouts as balloons; they don't follow BOM renumbering.
- Don't trust an HTTP 200 from a modify batch without the per-item check.
- Don't over-tolerance (±0.1 everywhere costs money).
- Don't dimension iso views or hidden lines, or double-dimension.
- Don't put cookies, XSRF values or tokens in scripts, headers, logs or chat.

## Common mistakes

- **No version before drawing,** so a later model edit silently moves views and dimensions.
- **Dangling annotations after an update,** left on the sheet. Always fix them after Ctrl+q.
- **Sheet-metal dimensions** without stating inside or outside mold line.
- **Wrong projection symbol** for the template (ISO first-angle sent to a US shop).
- **BOM quantities** that don't match the assembly because of suppressed or hidden instances.
- **Fit tolerances set via the API** without checking them in the GUI (undocumented).

## Checks before done

- [ ] Template and projection symbol are correct; size and scale are stated; units are stated.
- [ ] Title block complete from properties: PN, title, rev, material, finish, mass, general tolerance, drawn/checked, sheet n/m.
- [ ] Only the necessary views; internal features shown by section or detail; centermarks and centerlines present.
- [ ] Every feature is defined exactly once (try to "make" the part from the sheet).
- [ ] Hole callouts are linked; thread spec includes size, pitch, class and depth.
- [ ] Every tight tolerance has a calculation; datums match how the part mounts.
- [ ] Assembly: every BOM item ballooned once with linked balloons; torques and notes present.
- [ ] API-made: every batch item OK; re-exported JSON has no dangling items; PDF checked by eye.
- [ ] The document is versioned and the PDF/DXF exported from that version.

## Sources

- Drawings overview, create, views: https://cad.onshape.com/help/Content/Drawing/drawings.htm , https://cad.onshape.com/help/Content/Drawing/creating_a_drawing.htm , https://cad.onshape.com/help/Content/Drawing/views.htm
- Section / detail / auxiliary / break / broken-out / crop / aligned: https://cad.onshape.com/help/Content/Drawing/drawing_section_view.htm , https://cad.onshape.com/help/Content/Drawing/detail_view.htm , https://cad.onshape.com/help/Content/Drawing/auxiliary_view.htm , https://cad.onshape.com/help/Content/Drawing/break_view.htm , https://cad.onshape.com/help/Content/Drawing/broken_out_section_view.htm , https://cad.onshape.com/help/Content/Drawing/crop_view.htm , https://cad.onshape.com/help/Content/Drawing/aligned_section_view.htm
- Dimensions: https://cad.onshape.com/help/Content/Drawing/drawing_dimensions.htm , https://cad.onshape.com/help/Content/Drawing/drawing_ordinate_dimension.htm
- Hole callout / hole table: https://cad.onshape.com/help/Content/Drawing/hole_thread_callout.htm , https://cad.onshape.com/help/Content/Drawing/drawing_hole_table.htm
- GD&T / datum / surface finish / weld: https://cad.onshape.com/help/Content/Drawing/geometric_tolerance.htm , https://cad.onshape.com/help/Content/Drawing/datum.htm , https://cad.onshape.com/help/Content/Drawing/surface_finish.htm , https://cad.onshape.com/help/Content/Drawing/weld_symbol.htm
- BOM / balloons / revision table / notes: https://cad.onshape.com/help/Content/Drawing/bom_table.htm , https://cad.onshape.com/help/Content/Drawing/callout_balloon.htm , https://cad.onshape.com/help/Content/Drawing/revision_table.htm , https://cad.onshape.com/help/Content/Drawing/note.htm
- Templates and properties: https://cad.onshape.com/help/Content/Drawing/custom_drawing_templates.htm , https://cad.onshape.com/help/Content/Drawing/drawing_properties.htm
- Flat pattern view: https://cad.onshape.com/help/Content/Drawing/drawing_flat_pattern_view.htm
- Update / dangling / export: https://cad.onshape.com/help/Content/Drawing/updating_a_drawing.htm , https://cad.onshape.com/help/Content/Drawing/dangling_entities.htm ,

https://cad.onshape.com/help/Content/Drawing/exporting_a_drawing.htm

- Merging (drawings Keep/Replace only): https://cad.onshape.com/help/Content/Document/merging.htm
- Drawings API guide: https://onshape-public.github.io/docs/api-adv/drawings/ ; translations: https://onshape-public.github.io/docs/api-adv/translation/
- Standards (ASME Y14.5-2018, Y14.100, ISO 128/129-1/1101/2768/286/273/6410) are paywalled. The rules above are widely taught content, and clause-level detail is INFERENCE.
- Observed practice: the Onshape Explore "Jackhammer" drawings (ANSI title block, zones, hole table, section A-A, detail, iso 1:4, position FCF, toleranced diameters, chamfer callout, general notes, exploded assembly sheet with BOM and balloons) and its spec-named Hole features.

# onshape-quickstart-lessons-log

**Summary (description):** Use when opening Explore Onshape / Try Onshape / Quickstart tutorials or their sample documents (Bench Vise, Postman, Jackhammer) in the box browser, when a tutorial panel, iframe, new tab or dropped connection gets in the way, or when you want the real-design takeaway of each Quickstart lesson (overview, model a part, hole across multiple parts, build an assembly, detail a drawing) without redoing it.

# Onshape Quickstart / Explore Onshape lessons log

> **Native features only, never FeatureScript.** The samples contain custom features (e.g. Postman's "Laser Joint", "Parametric Part Numbers"); study them, never copy or insert them. Team documents are read-only; do exercises in a personal doc named `Learning - <lesson>`.

Labels: FACT = seen in Explore Onshape / help pages; INFERENCE; GUESS. Lesson detail below is a summary.

## 1. Where things are (FACT)

- Explore Onshape: `the Onshape "Explore Onshape" page` (lightbulb "Explore Onshape" button top-right in any document). Its content is an **iframe**; cards open in the same tab and have no href.
- Sections: **Try Onshape** Quickstart (1 Onshape overview, 2 Model a part, 3 Build an assembly, 4 Detail a drawing), **Sample models** (Sample assignment, Postman, Jackhammer), Teach with Onshape (CAD Basics, Onshape Fundamentals, Education Enterprise, Teaching a class), and ~20 Learning Center pathways (Part modeling: fundamentals, top-down, part configurations, variables and expressions, master model, advanced part design, surfacing; Assemblies: intro, configurations, advanced, managed in-context; Advanced: API, PCB studio, simulation).
- Learning Center pages are JavaScript-only: curl/WebFetch get an empty shell; lesson pages show "Purchase Required" without sign-in. Use the box browser signed in (onshape-gui-fast-recipes).
- Each Quickstart step is a video + transcript in a side panel; advance with the panel's Next; a Check button validates exercise steps. Progress shows as a Quickstart % (overview complete = 38%).

## 2. Lesson takeaways that transfer to real design

| Lesson | Takeaway for real parts |
|---|---|
| Onshape overview (done, FACT) | Document = project container (Part Studios, Assemblies, Drawings, imported files) with one versioned history, live sharing, no file locks. One Part Studio history can define several parts (top-down). Selection is persistent: no Ctrl needed to multi-select; Space or empty click clears. Put a bracket's studio, assembly and drawing in one doc so a hole change flows to mates and drawing. |
| Model a part (partly done) | Steps: Edit a sketch, Create new hole geometry, Learn how changes are tracked. Sketch on a plane → fully define (black) → extrude → Hole feature → fillets last. Name features. |
| Hole across multiple parts (Model a part Step 2, FACT) | Hole feature from the "USE ME!" sketch through JAW, STATIC and SHEET METAL PLATE: one Hole, several parts in Merge scope, Tapped + Fastener fit gives clearance in the first parts and tapped in the last (onshape-hole-feature-and-standard-content §2). |
| Build an assembly (INFERENCE) | Insert from the Part Studio, **fix one** ground part, mate by real DOF (Fastened, Revolute, Slider), then standard-content screws. |
| Detail a drawing (INFERENCE) | Views from the part, Hole/Thread callouts (work only on Hole-feature holes), title block from properties, BOM + balloons on assembly drawings. |

## 3. What the sample documents teach (FACT from earlier notes)

- **Postman** (VEX-scale robot): `PS/A` numbering per subsystem, one frame Part Studio driven by a reference sketch, build-right-mirror-left, Derived master sketches in the Arm studio, top assembly only Fastened mates, assembly configurations for wheel variants.
- **Jackhammer**: variables folder at the top, layout sketch separate, holes named by spec (`M8x1.25 ↧ 35 mm`), configured composite ball bearing (6001/6201/... as configuration names), assembly drawing = exploded view + BOM + balloons, part drawing = GD&T, hole table, toleranced diameters.
- **FRC 6329** (Student Team Examples): one top-level assembly of version-pinned subsystem documents with only Fastened mates; it uses an Origin Cube (FeatureScript) — replace with a native datum Part Studio (onshape-top-down-master-layout).

## 4. Box-browser gotchas (FACT, observed)

- **Bench Vise sample opens a new tab.** Clicking "Onshape overview" opens the lesson's own "Bench Vise" document in a new tab; work happens there. Switch tabs deliberately; close extra copies when done.
- **Pop-ups**: "Open linked document" also opens a new tab and Chrome may block it; allow from the omnibox icon.
- **Iframe panels**: automation must target the frame; the panel can sit over its own Next button (forced click). Keyboard shortcuts typed while the panel has focus don't reach the model, and text meant for a dialog can trigger shortcuts if the dialog isn't focused. Click into the field first.
- **Hole tool**: opening it via Search tools + Enter did nothing; clicking the Hole toolbar icon works. First open can hang on "Preparing #featurename…": reload, the dialog comes back.
- **Connection drops**: the box browser often shows "Onshape is not connected… Attempting to reconnect" or ERR_TIMED_OUT, worse when the user's computer is offline (traffic goes over the box network). After each drop the last action may not have saved: wait up to 60 s, reload, verify, then continue. Heavy API bursts get HTTP 429; stop, don't script around it (onshape-api-rate-limits).
- **Big samples load slowly** (Jackhammer assembly): open Part Studios first, use simplified display states.
- Linked documents in samples may fail with "Resource does not exist, or you do not have permission" — expected.

## 5. Worked mini-example: turn the lessons into a real part in 15 minutes (INFERENCE)

Goal: an FRC motor-mount bracket using only what the Quickstart taught. 1. New doc `Learning - bracket`. Variables: `#t = 0.125 in`, `#boltC = 1.0 in`. 2. Model a part: L-bracket sketch, extrude; Hole feature #10 Clearance Free at layout points; fillet edges last. 3. Hole across multiple parts: add a test block; one Tapped #10-32 Hole through bracket + block, block last. 4. Build an assembly: insert both, fix the block, Fastened mate, standard-content #10-32 SHCS by face selection. 5. Detail a drawing: front + side + iso, Hole/Thread callouts, material 6061, title block. 6. Self-check with mech-design-bolted-joints-and-fasteners (edge distance >= 1.5–2x D, screw length).

## 6. Open items

- Learning Center lesson bodies (Managed In-Context, Master Model, Simulation, Assembly configurations) need sign-in; only outlines and video summaries were read.
- Postman Intake, Jackhammer Render Studio and Manual, Sample assignment, Teach with Onshape — not studied yet.

## 7. Done when (after any Explore/Quickstart session)

- [ ] Extra tabs (Bench Vise copies, linked docs) closed; no stray personal copies of team docs.
- [ ] Each takeaway written into the matching skill, not left in chat.

- [ ] No FeatureScript copied from a sample; native replacement noted instead.
- [ ] This log updated with lesson status and new gotchas.

# onshape-release-and-versioning

**Summary (description):** Use whenever you freeze, share, link, branch, merge, protect or release Onshape design data — naming versions, versioning before links/derives/handoffs, updating linked references, experimenting on branches, comparing and merging, workspace protections, publications, and release candidates with revisions.

# Onshape versions, branches and release

> **Native features only.** Never use FeatureScript: no custom features, Feature Studios, `/featurescript` or evalFeaturescript. Version control here uses only built-in Versions and history, Branches, Compare, Merge, Workspace protections, Publications and Release management. Never create versions, branches or releases in documents you don't own or weren't asked to change.

## Core facts

- A **version** is an immutable, view-only snapshot. Anything defined in a version is stable and recoverable.
- **Linked documents always reference versions.** In the using document you decide whether and when to update to a newer version. Inside one document, changes propagate instantly.
- **Derived** features also link by version across documents.
- When you version, Onshape offers to update drawings that have pending updates, so the version holds current drawings.
- Onshape may create a version automatically when you update document references.
- **Merge** is per tab (Keep, Merge or Replace). Only Part Studios and Assemblies merge; **Drawings and PCB Studios can only be Kept or Replaced.**
- **Workspace protections:**
- Geometry, drawing, some metadata, and tab create/delete changes must be made on a branch and merged.
- Release, drawing refresh, versioning, branching and merging are still allowed directly.
- A protected workspace is *not* read-only.
- **Publications** are view-only packages of versions or revisions from many documents, shareable with any Onshape user.
- **Release management** (plan-gated; availability on Education/Free plans UNVERIFIED):
- Release candidates give revisions to parts, assemblies and drawings.
- Configurations can be released individually, and old revisions can be obsoleted.

## Workflow

1. **Name versions so a stranger understands them.** Pattern (INFERENCE): `V<major>.<minor> <subsystem> <what changed>`, e.g. `V2.3 Intake — wider rollers, 1/8 plates`. Fill the description with the why and the open issues.
2. **Version at every handoff:**
3. before linking or deriving into another document;
4. before drawings and exports go out;
5. before CAM or a manufacturing release;
6. before an event or design review;
7. before any batch or API change;
8. before a risky refactor.

9. **Link, never copy.** Insert parts and subassemblies from another document's version. Record the source version in the assembly instance name or a note when it matters.
10. **Update references on purpose.**
11. Open the update dialog (the update icon on the instance list or Part Studio).
12. Read which documents have newer versions.
13. Update one subsystem at a time, then check mates, derived geometry and drawings.
14. Version the using document after the update.
15. **Experiment on a branch.**
16. Versions and history → branch from a version, with a name like `exp-claw-v2`.
17. Make the changes, version the branch, then **Compare** branch vs main (geometry, features, properties).
18. **Merge back.**
19. From main: Merge → pick the branch's version → per tab choose Merge, Keep or Replace.
20. Drawings: decide Keep or Replace, since they can't merge. Keep drawing work on one branch to avoid losing it.
21. Version main right after the merge.
22. **Protect main** when a team works in parallel: right-click Main → Workspace protections. All edits then go branch → merge, which works like code review.
23. **Restore** an earlier state from Versions and history (restore a tab or the workspace to a version) instead of hand-undoing many features.
24. **Share read-only** to sponsors, vendors or other teams with a **Publication** (only versions or revisions can be inserted) or a view-only link to a version. Never share the edit workspace.
25. **Release (when the plan has it):**
    - Open a Release candidate from the Parts list, a tab right-click, Properties, or Versions and history.
    - Add items with +; tick **Include associated drawings**.
    - Set the revision scheme; **require an approver who isn't the creator**.
    - Released revisions show in revision tables on drawings.
    - Obsolete superseded revisions.
26. **Without release management** (typical team plans): use version names plus a manual revision table (General table) on drawings, and a configured or typed Revision property.
27. **Folder permissions:** keep the root folder minimal and grant access per document or publication.

## Version naming and cadence table (INFERENCE, team practice)

| Moment | Version name example | Also do |
|---|---|---|
| Layout frozen for subsystem start | `V1.0 Layout — frame perimeter locked` | Tell the subsystem owners to link to it |
| Subsystem design review | `V1.4 Arm — CDR` | Attach review notes in the description |
| Sent to manufacture | `V1.5 Arm — MFG plates rev A` | Export PDF/DXF from this version |
| After robot assembly changes | `V2.0 Robot — comp 1 build` | Update the top assembly references |
| Pre-event freeze | `V2.3 Robot — District 1 freeze` | Print drawings and the BOM from it |

## Pre-event freeze recipe (INFERENCE, team practice)

1. Ask every subsystem owner to version their document with a `… — <event> freeze` name.
2. In the robot document, update the references to exactly those versions. Rebuild, then check mates, configurations and any interference check.
3. Fix breaks on a branch (`fix-<event>`), compare, merge, and version main as `V<x> Robot — <event> freeze`.
4. From that version: export the drawing PDFs, the flattened BOM (CSV) and DXFs for spare plates, plus a spare-parts list.
5. Put the version link in the pit binder or tech binder so anyone can open the exact CAD at the event.
6. After the event, record what was changed on the robot and model it on a branch (`as-built-<event>`) before merging.

## Do

- Version before every link, derive, export, release, review and event.
- Update linked references deliberately, one subsystem at a time.
- Use branches for alternatives; use configurations only for variants that must coexist (see `onshape-variables-and-configurations`).
- Compare before you merge; version right after.
- Keep drawings on a single branch line.

## Don't

- Don't make copies of documents to "save" states; use versions.
- Don't link to workspaces for anything others depend on; link to versions.
- Don't merge drawings expecting a combination; they're Keep or Replace only.
- Don't self-approve releases.
- Don't create versions, branches or releases in documents you don't own.

## Common mistakes

- **Vague names** like "v3" or "fixed stuff," so nobody can tell which version was cut.
- **Forgetting to update references** after a subsystem change, so the robot assembly shows old parts. Or **updating everything at once** and breaking many mates without knowing which change did it.
- **Long-lived branches** that diverge until the merge is painful. Merge or drop within days.
- **Exporting DXFs from the workspace** instead of a version, so the shop's file doesn't match any saved state.
- **Drawing edits on two branches,** so one set is lost on Replace.
- **Assuming release management exists** on the team's plan without checking.

## Checks before done

- [ ] Branch names follow a pattern (`exp-…`, `fix-…`, `as-built-…`), and stale branches are noted in their version descriptions.
- [ ] Where used was checked before changing a shared part (to see which documents link to it).
- [ ] The version has a clear name and description.
- [ ] Every downstream document links to versions, and the update status was reviewed.
- [ ] Branches were compared before merging, and main was versioned after the merge.
- [ ] Drawings and exports come from a named version.
- [ ] Sharing used a publication or a view-only version link.
- [ ] Releases have a non-creator approver and include associated drawings.
- [ ] No version-control actions in documents you don't own.

## Sources

- Versions and history (create, branch, restore, automatic versions, drawing updates): https://cad.onshape.com/help/Content/Document/versions_and_history.htm
- Linking documents (references to versions; you control updates): https://cad.onshape.com/help/Content/Document/linking_documents.htm
- Compare: https://cad.onshape.com/help/Content/Document/compare.htm ; Merging (drawings/PCB Keep or Replace): https://cad.onshape.com/help/Content/Document/merging.htm
- Workspace protections: https://cad.onshape.com/help/Content/Document/workspace_protections.htm
- Restoring: https://cad.onshape.com/help/Content/Document/restoring.htm ; Where used: https://cad.onshape.com/help/Content/Document/where_used.htm

- Publications: https://cad.onshape.com/help/Content/Document/publications.htm ; Folders: https://cad.onshape.com/help/Content/Document/folders.htm
- Derived: https://cad.onshape.com/help/Content/PartStudio/derived.htm
- Release: https://cad.onshape.com/help/Content/Release/release_management.htm , https://cad.onshape.com/help/Content/Release/typical_release_workflow.htm , https://cad.onshape.com/help/Content/Release/releasing_a_configuration.htm , https://cad.onshape.com/help/Content/Release/viewing_revision_history_and_obsoleting_parts.htm , https://cad.onshape.com/help/Content/Release/specifying_approvers.htm
- Revision table on drawings: https://cad.onshape.com/help/Content/Drawing/revision_table.htm
- Plan availability of release management: UNVERIFIED (not stated on the help pages read).

# onshape-sheet-metal

**Summary (description):** Use when designing any bent plate, gusset, bracket, enclosure or rolled part in Onshape with the native Sheet metal model, including vendor-matched thickness, bend radius and K-factor, flanges, reliefs, corner breaks, flat patterns, DXF export and sheet-metal drawings.

# Onshape sheet metal

> **Native features only.** Never use FeatureScript: no custom features (including "Laser Joint", tab-and-slot or "Lighten" generators), Feature Studios, `/featurescript` or evalFeaturescript. Use the built-in Sheet metal model, Flange, Hem, Tab, Joint, Bend, Corner, Bend relief, Corner break, Form and Finish features, the way a person would.

## When to use

- Bent aluminum or steel parts: gussets, bellypans, brackets, intake plates bent into channels, electronics trays, covers.
- Flat laser or waterjet plates that may later get bends.
- Rolled shapes (extruded arcs or splines).

## Before modeling: get the vendor numbers

| Parameter | Where it comes from | Example (INFERENCE, confirm with your vendor) |
|---|---|---|
| Thickness | Stock list | 0.090 in 5052-H32 |
| Inside bend radius | Press-brake tooling | ≈ 1× thickness for 5052 |
| K-factor | Vendor or bend test | Onshape default 0.45; rolled default 0.5 |
| Minimal gap (rips) | Cutter kerf and clearance | 0.010–0.020 in |
| Relief type and scale | Shop preference | Rectangle or round, scaled |
| Min flange length | Tooling die opening | Often ≥ about 4× thickness plus radius |

## Workflow

1. **Create the model** with Sheet metal model, choosing the operation:
2. **Convert:** select a solid; pick the faces to exclude and the edges that become **bends** (unselected edges become **rips**; unselected arcs or splines become tangent joints).
3. **Extrude:** from a sketch; arcs and splines make rolled walls.
4. **Thicken:** from faces or sketches.
5. **Set the General parameters:**
6. Thickness and its direction; inside bend radius;
7. bend calculation: K Factor (default), Bend allowance or Bend deduction;
8. K-factor; rolled K-factor; minimal gap;
9. corner relief type and scale; bend relief type, with relief width = thickness × scale.
10. **Open the flat view and the sheet metal table side by side.** Folded, flat and table update together.
11. In the table: convert rip ↔ bend, change joint types, edit bend radii.
12. **Add walls with Flange:**
13. Alignment Inner, Outer or Middle; end type Blind, Up to entity, or Up to entity with offset; angle control;
14. automatic miter (default on); "use model bend radius" (default on); partial flange with offsets when needed.

15. **Other walls and edges:** Hem (safe edges), Tab (added material on a wall), SM Loft (transitions), Make joint, Corner (corner style and gap), Bend (bend along a line in the flat).
16. **Cuts and holes:**
17. Use Extrude remove or the Hole feature while the model is active. Cuts go perpendicular to the walls, so they flatten correctly.
18. Lightening pockets: sketch them natively, extrude remove, and pattern the feature. No "Lighten" custom feature.
19. **Tab-and-slot joints between plates (the native replacement for "Laser Joint"):**
20. Sketch the tab profile on the mating edge; Extrude add on one part.
21. Extrude remove the slot (tab + clearance, e.g. +0.005–0.010 in per side, INFERENCE) through the other part.
22. Feature-pattern it along the edge. Name the features (`Tabs Side-to-Bottom`, `Slots Bottom`).
23. **Forms:** use Sheet metal form with library forms or custom forms. Model all of a custom form's detail when you create the form, not later.
24. **Corner break last,** after flanges and joints are final. Once applied, the joint type and style can't be changed.
25. Fillet (radius or width) or chamfer on the corner edges or vertices, in the folded or flat state.
26. **Finish sheet metal model** only for operations done *after* fabrication (angled drilling, a welded corner). After Finish, features act on the 3D solid.
27. **Material and properties:** set the material for mass; set the part number and stock thickness in properties.
28. **Flat pattern output:**
    - Drawing: insert a **Flat pattern view** from the Insert dialog. Up and down bend lines display with their own line style.
    - Add a bend note or table (direction, angle, radius).
    - DXF/DWG for the cutter: export the flat pattern. Confirm the menu path in the current UI; it is offered from the sheet-metal flat view or part export.

## Hand check of the flat length (do this once per new vendor setting)

- Bend allowance: BA = θ(rad) · (R + K·T).
- Outside setback: OSSB = (R + T) · tan(θ/2).
- Bend deduction: BD = 2·OSSB − BA.
- Flat length from outside leg dimensions A and B: L_flat = A + B − BD.
- **Worked example** (T = 0.090 in, R = 0.090 in, K = 0.45, θ = 90°, outside legs A = 1.000 in and B = 2.000 in):
- BA = 1.5708 × (0.090 + 0.0405) = **0.2050 in**
- OSSB = 0.180 × tan 45° = 0.1800 in
- BD = 0.3600 − 0.2050 = **0.1550 in**
- L_flat = 3.000 − 0.1550 = **2.8450 in**
- **Pass** if Onshape's flat-pattern length (Measure in the flat view) matches within ±0.002 in. **Fail** means the K-factor, radius or leg reference (inside vs. outside) differs. Fix the model parameters, not the DXF.
- Then compare with the vendor's own bend-deduction chart for that tooling. If they differ, use the vendor value (switch the bend calculation to Bend deduction and enter it).
- Formulas: standard bend-allowance relations (general sheet-metal practice). The example numbers are computed here.

## Sheet-metal drawing minimum

1. A folded view set: front, top, and an iso for orientation.

2. A flat pattern view (Insert dialog), with overall flat size, bend lines, and UP/DOWN direction, angle and inside radius per bend.
3. Notes: material and temper, thickness, "break all sharp edges" or a deburr note, and the K-factor or bend-deduction basis.
4. The hole table on the flat if the plate is hole-heavy; see `onshape-professional-drawings`.
5. Version the document before you send the DXF and PDF together, so both match one version.
6. File names carry the part number and version, e.g. `PN-1234_v3_flat.dxf`.

## Do

- Match the K-factor and bend radius to the real shop before sending files.
- Keep one sheet-metal model per part family; multiple active models are allowed.
- Dimension flanges from the outside mold line or inside, consistently with how the vendor measures.
- Check the flat for overlaps and interference (the flat view shows them).
- Use bend reliefs wherever a bend meets an edge without a relief.

## Don't

- Don't use the solid Fillet or Chamfer on sheet metal; use Corner break.
- Don't add corner breaks before the flanges and joints are final.
- Don't Finish the model just to add normal holes; holes go in while it is active.
- Don't use custom features for tabs, slots, lightening or joints.
- Don't send a flat pattern without checking the thickness, units and bend count against the drawing.

## Common mistakes

- **The default K 0.45 left in** when the vendor's is different, so the bent flange lengths come out wrong. Set it per vendor.
- **Holes too close to a bend** (distorted after bending). INFERENCE: keep holes ≥ about 2× thickness plus the radius from the bend line.
- **Flanges shorter than the tooling allows.** Check the minimum flange length with the shop.
- **Missing reliefs:** tearing at the bend ends.
- **Rips with zero gap:** the flat overlaps. Set a minimal gap.
- **A DXF exported in the wrong units** or with bend lines on the cut layer. Check it in a viewer before ordering.

## Checks before done

- [ ] Thickness, bend radius, K-factor and relief match the vendor sheet.
- [ ] The flat view has no overlaps; the table shows the expected bends and rips.
- [ ] Corner breaks are the last sheet-metal features; no solid fillets or chamfers on the part.
- [ ] Material is set; the mass is reasonable.
- [ ] The flat-pattern drawing view has bend directions and notes.
- [ ] The DXF is checked for units and scale against one known dimension.
- [ ] No custom features in the feature list.

## Sources

- Sheet metal model (operations, K-factor 0.45 default, rolled 0.5, bend calculation, minimal gap, reliefs, Finish): https://cad.onshape.com/help/Content/PartStudio/sheet_metal_model.htm
- Flange: https://cad.onshape.com/help/Content/PartStudio/sheet_metal_flange.htm ; Hem: https://cad.onshape.com/help/Content/PartStudio/sheet_metal_hem.htm ; Tab:

https://cad.onshape.com/help/Content/PartStudio/sheet_metal_tab.htm

- Joint / Make joint / Corner / Bend relief / Bend: https://cad.onshape.com/help/Content/PartStudio/sheet_metal_joint.htm , https://cad.onshape.com/help/Content/PartStudio/sheet_metal_make_joint.htm , https://cad.onshape.com/help/Content/PartStudio/sheet_metal_corner.htm , https://cad.onshape.com/help/Content/PartStudio/sheet_metal_bend_relief.htm , https://cad.onshape.com/help/Content/PartStudio/sheet_metal_bend.htm
- Corner break (last; locks joint type): https://cad.onshape.com/help/Content/PartStudio/sheet_metal_corner_break.htm
- Form and form library: https://cad.onshape.com/help/Content/PartStudio/sheet_metal_form.htm , https://cad.onshape.com/help/Content/Library/sheet_metal_form_library.htm
- Finish: https://cad.onshape.com/help/Content/PartStudio/finish_sheet_metal_model.htm ; table: https://cad.onshape.com/help/Content/PartStudio/sheet_metal_table.htm
- Flat pattern drawing view: https://cad.onshape.com/help/Content/Drawing/drawing_flat_pattern_view.htm
- Export formats: https://cad.onshape.com/help/Content/File/exporting_files.htm
- Fillet / Chamfer warnings for sheet metal: https://cad.onshape.com/help/Content/PartStudio/fillet.htm , https://cad.onshape.com/help/Content/PartStudio/chamfer.htm

# onshape-sketching-and-constraints

> **Summary (description):** Use whenever you create or edit an Onshape sketch (part profiles, hole-point sketches, layout/master sketches, sweep paths, imported DXF outlines) and need it fully defined, robust to change and driven by variables.

# Onshape sketching and constraints

> **Native features only.** Never use FeatureScript: no custom features, Feature Studios, `/featurescript` or evalFeaturescript, and no "Any"-type variables that carry FeatureScript values. Sketch with the built-in tools and constraints, the way a person would.

## When to use

- Any new sketch: profile, hole-location sketch, layout sketch, path, or reference sketch.
- A sketch went blue or red, or downstream features moved unexpectedly after an edit.
- Importing a DXF/DWG outline (vendor flat, field drawing, plate pattern).

## Workflow

1. **Pick a stable sketch plane.**
2. Use a default plane (Top, Front, Right), a named reference plane (`Frame Top Plane`, `Wheel Axis Plane`), or a flat face of a *base* feature.
3. Never use faces made by fillets, chamfers, late detail cuts or pattern instances.
4. **Name the sketch for its job** (Shift+n): `Side Plate Profile`, `Bolt Hole Locations Top Plate`, `Top View Drivetrain Reference Sketch`.
5. The Explore models name sketches by function. Postman's Part Studios hold `Top View … Reference Sketch`, `Centered reference sketch` and `Bolt Hole Locations Top Plate`.
6. **Bring in references before drawing.**
7. Use `Use` (u) on layout or derived sketch geometry, or on stable edges. Projected references show a blue background on their constraint.
8. Prefer referencing the **layout sketch** over another part's edges.
9. **Rough in the geometry at about the right size and shape**, then constrain it. Close to final size keeps the solver from flipping.
10. **Add construction geometry (q):**
11. centerlines, symmetry axes, bolt circles, pitch circles, belt or chain lines, range-of-motion arcs.
12. **Add geometric constraints *before* dimensions** (shortcuts): | Constraint | Key | Typical use | |---|---|---| | Coincident | i | Snap endpoints to references or points | | Horizontal / Vertical | h / v | Datum-aligned edges, hole rows | | Equal | e | Repeated slot widths, matching holes | | Parallel | b | Plate edges | | Perpendicular | Shift+l | Square corners | | Tangent | t | Slots, arcs into lines | | Concentric | Shift+o | Bearing bores to shaft axes | | Midpoint | Shift+m | Centered features | | Symmetric | Shift+q | Mirror-image profiles about a construction line | | Pierce | Shift+g | Sweep profile or layout to a 3D curve | | Normal | Shift+k | Lines normal to curves | | Fix | – | Only for imported geometry you intentionally freeze |
13. **Add dimensions (d) last.**
14. Dimension to datums and references, not to other dimensioned geometry in a chain.
15. Type expressions with units and variables: `#frame_w / 2 - 0.5 in`, `2 * #wall`, `#bolt_pcd / 2`.
16. **Drive it to black.**
17. Blue = under-defined, black = fully defined, red = over-defined/conflict.

18. Drag any blue entity to find what is still free. Delete redundant constraints rather than adding more when anything shows red.
19. **Keep profiles simple.**
20. One profile intent per sketch.
21. Let features do fillets, patterns and mirrors. A sketch fillet is fine only when the profile itself needs it, e.g. a slot end.
22. Hole sketches: points only. The Hole feature makes the circles.
23. **Exit and check:** the region count, no zero-length or overlapping entities, the feature status is OK, and the sketch is fully defined.

## Numeric-field rules

- **Units:** mm, cm, m, in, ft, yd, deg, rad. Mixing is fine when every term has a unit (`25 mm + 1 in`).
- **Operators and functions:** + − * / ^ %, sqrt, abs, min, max, ceil, floor, round, exp, log, log10, atan2, pi. Trig uses degrees.
- **Invalid:**
- `3 + 2.5 in` (unitless plus length);
- `(2*3)(1/3)` (implicit multiplication; write `*`).
- **Variables:** `#name`. They must be defined *above* this feature in the list, or in a Variable Studio that has been inserted.
- **Scrolling a field:** wheel steps 0.1; Ctrl+wheel 0.01; Shift+wheel 1.0.
- **Hand-math values** (belt C-C, chain C-C, gear center distance; see `engineering-math-proofs`): compute them in a Variable feature as a plain expression, e.g. for equal HTD pulleys `#cc = (#belt_teeth - #pulley_teeth) / 2 * 5 mm`, or type the checked number.
- For unequal pulleys, solve the exact formula in the proof script and paste the result into a Length variable with a comment naming the script. **Never** use custom FeatureScript functions such as `#BeltCTC_5mm(...)` from a Feature Studio library.

## Layout and master sketches (special rules)

- Keep them in a dedicated layout Part Studio (see `onshape-top-down-master-layout`) and constrain them to the origin and default planes only.
- 6329 measured its 2024 layout to the robot origin, not to the bottom of the drive tubes.
- Use construction geometry for envelopes and motion arcs. Use solid lines only for what downstream parts will `Use`.
- Add a sketch per robot state, or arcs showing the range of motion, so poses can be checked against height and extension limits.
- Put key points (pivots, axle centers, belt pulley centers) as sketch points. Downstream sketches `Use` the points and Mate connectors snap to them.

## Special sketches

- **Sweep path and profile:** sketch the profile on a plane normal to the path's end, and constrain it to the path with Pierce or Coincident.
- **Imported DXF/DWG:** insert it into an *empty* sketch. Check the units and scale with one known dimension, then constrain or Fix it. Do not mix imported and new geometry in the same sketch.
- **Image tracing:** Insert image, scale it with a known dimension, trace in a separate sketch, then hide the image.
- **Text:** Sketch text for engraving or labels. Keep it shallow, and away from thin walls.
- **Sketch patterns:** Linear and Circular sketch patterns are fine for repeated *sketch* points (hole points). For repeated solid features, use feature patterns.

## Do

- Constrain to the origin or reference geometry first, so the sketch can't float.
- Use Symmetric about a construction centerline rather than two equal dimensions.
- Use Equal across repeated features so one dimension drives them all.
- Name the driving dimensions via variables when other features or parts depend on them.
- Use `Use` (u) references to layout geometry, and keep the chain one level deep: layout → part sketch.
- Use Intersection (sketch tool) or Use on silhouettes only when the source is stable.

## Don't

- Don't leave blue (under-defined) sketches in released work.
- Don't Fix entities to "make it black". Fix hides missing intent; reserve it for imported geometry.
- Don't dimension a closed chain completely: it over-defines (red). Leave one dimension driven, or use a reference.
- Don't reference edges of fillets, chamfers, patterned instances or another part's detail features.
- Don't draw everything in one huge sketch. Split by purpose: profile, holes, cuts.
- Don't use FeatureScript functions in expressions.

## Common mistakes

- **Downstream breakage after an edit.** It was referencing a face that was regenerated. Re-attach to a named plane or the layout.
- **Solver flips** (an arc goes the wrong way, a line crosses). The geometry started too far from its final size. Undo, drag closer, then dimension.
- **Over-defined red after `Use`.** The projected entity was already constrained. Delete the duplicate constraint.
- **Units wrong after a DXF import** (mm read as in). Check one known length immediately.
- **A sweep profile not on the path plane.** The sweep fails or twists. Put it on a normal plane and Pierce it.
- **Equal applied across mismatched entity types,** e.g. an arc and a line, which gives unexpected results.

## Checks before done

- [ ] The sketch is black: no blue or red entities, no warnings.
- [ ] It is named for its function. It sits on a named plane or a stable base face.
- [ ] References come from the layout or stable geometry only, one level deep.
- [ ] Driving values are variables or expressions with units; there are no FeatureScript functions.
- [ ] Hole sketches contain points only (for Hole features).
- [ ] Downstream features still regenerate after a ±10 % change to the key variables. Test it, then undo.

## Sources

- Constraints, colors, Use: https://cad.onshape.com/help/Content/Sketch/working_with_constraints.htm , https://cad.onshape.com/help/Content/Sketch/use.htm , https://cad.onshape.com/help/Content/Sketch/sketch_basics.htm
- Individual constraints: https://cad.onshape.com/help/Content/Sketch/pierce.htm , https://cad.onshape.com/help/Content/Sketch/symmetric.htm , https://cad.onshape.com/help/Content/Sketch/fix.htm
- Construction: https://cad.onshape.com/help/Content/Sketch/construction.htm ; troubleshooting: https://cad.onshape.com/help/Content/Sketch/troubleshooting_sketch_geometry.htm

- DXF/DWG: https://cad.onshape.com/help/Content/Sketch/insert_dwg_or_dxf.htm ; image: https://cad.onshape.com/help/Content/Sketch/insert_image.htm ; text: https://cad.onshape.com/help/Content/Sketch/text.htm
- Numeric fields: https://cad.onshape.com/help/Content/Home/numeric_fields.htm ; Variables: https://cad.onshape.com/help/Content/PartStudio/variable.htm
- Sweep: https://cad.onshape.com/help/Content/PartStudio/sweep.htm
- Keyboard shortcuts: https://cad.onshape.com/help/Content/Home/keyboard_shortcuts_and_hotkeys.htm
- 6329 layout measured to the origin (Drew_B): https://www.chiefdelphi.com/t/team-6329-the-bucks-wrath-2024-robot-reveal-dial/456061
- Observed practice: the Onshape Explore "Postman" sample, with its reference and hole-location sketches named by function.

# onshape-surfacing

**Summary (description):** Use when a part needs freeform or blended shape that extrude/revolve can't make — covers, shrouds, ducts, ergonomic handles, bumper corners, game-piece guides — or when repairing/editing imported geometry (STEP/mesh) in an Onshape Part Studio; low priority for typical FRC plate-and-tube work.

# Onshape surfacing

> **Native features only.** Use the built-in surface, direct-edit and analysis tools. Never use FeatureScript, custom features, Feature Studios or the FS API endpoint, even for "smooth surface" or lofting helpers.

## When to use

- The shape isn't prismatic: a curved polycarbonate cover, a 3D-printed duct, an intake funnel, a handle grip.
- An imported model has gaps or bad faces that need repair, or needs edits without its history (direct editing).
- Not for flat plates, tubes or sheet-metal brackets. Use Extrude, Frame or Sheet metal instead.

## Core facts

- Surfaces have **no mass**. Convert them to solids with **Thicken** (add depth: New, Add, Remove or Intersect) or **Enclose** (all boundaries around an empty volume must intersect or connect at a boundary).
- **Continuity:**

| Level | Meaning | Set with |
|---|---|---|
| G0 | position | any shared edge |
| G1 | tangent | "Match" in Bridging curve, Fillet, Face blend |
| G2 | curvature | "Match" curvature options |
| G3 | flow | rarely needed in FRC |

- **Surface choice:**

| Feature | Use | Note |
|---|---|---|
| Extrude, Revolve, Sweep (surface option) | simple surfaces | most robust |
| Loft | between profiles, with guides | 4-sided control, high quality |
| Boundary surface | U and V curve networks, with boundary conditions (tangency, normal to profile) | four-sided, highest control |
| Fill | patching N-sided holes or gaps | lower quality than Loft or Boundary; good for repairs |
| Ruled surface | from edges: Normal, Tangent, Aligned with direction, Angle from direction | draft walls, runoff faces |
| Offset surface | a parallel copy of faces | shells, clearances |
| Constrained surface | from points or mesh within a tolerance | scans/meshes; created on desktop or iOS only |

- **Direct editing** (no history needed): Move face (Offset, Translate, Rotate), Replace face, Delete face (heal, cap or leave open), Split (part or face by a plane, surface or sketch).

## Workflow

1. **Define the envelope first.** Derive the layout sketch and name the planes. Sketch the key section curves fully constrained, with tangent or curvature constraints where the curves join.
2. **Build clean curves.** Use sketch splines with few control points, Bridging curve (Match G1/G2) between edges, and native curves.
3. **Don't build from trimmed edges or projected curves:** they carry many cubic spans, so whatever is built from them is "more complex and imperfect" (Onshape help).
4. **Make the primary surfaces.** Use the simplest tool that works: Extrude or Revolve the surface, then Sweep, then Loft, then Boundary surface. Keep patches roughly four-sided.
5. **Blend and trim.** Use Face blend or Fillet with G1/G2 match between surfaces. Use Split, or Boolean on surfaces, to trim. Extend or Move face (offset) where patches must overlap.
6. **Patch gaps** with Fill only where four-sided tools can't reach (N-sided holes), choosing boundary conditions per edge.
7. **Analyze** (Show analysis tools, bottom right):
8. **Curve/surface analysis** (Shift+c): combs should rise and fall smoothly; a sparse control-point grid is cleaner.
9. **Connection analysis:** G0/G1/G2 across edges.
10. **Zebra stripes:** stripes line up across a G1 edge and flow smoothly across G2.
11. **Curvature color map:** look for unexpected hot spots or sign flips.
12. **Reflection analysis:** an environment map shows visual quality.
13. **Dihedral analysis** (Shift+d): the angle across an edge; ~180° means tangent.
14. **Deviation analysis:** minimum and maximum distance between two curve chains or two surfaces, e.g. against a scan or reference.
15. **Draft analysis:** for molded or printed parts.
16. **Thickness analysis:** after thickening, shows thin spots.
17. **Highlight boundary edges** (Camera and render options): red lines show laminar or open edges that must be closed before Enclose.
18. **Make it solid.** Thicken the surface for shells and covers, choosing the inward or outward direction. Enclose for closed volumes (Keep tools if the surfaces are needed later).
19. **Finish:**
20. Fillets last, materials assigned.
21. Check mass in Mass properties.
22. Run Thickness analysis against process minimums: print wall ≥ 3–4 nozzle widths; thermoformed polycarbonate thickness is GUESS until it is tested.
23. **Manufacture:**
24. Export STL or 3MF for printing, or Parasolid or STEP for vendors.
25. A developable cover (single-curvature) can be laser cut flat and bent. Check Gaussian curvature ≈ 0 with the curvature map.

## Repairing imported or mesh models

- Turn on Highlight boundary edges to find gaps.
- Use Delete face (heal) for small bad faces, Fill to patch holes, then Enclose to get a solid.
- Use Move face or Replace face to fix positions without history.
- For meshes, model solids around them, or use a Constrained surface from mesh within a tolerance and check it with Deviation analysis.
- Exchange with vendors in Parasolid where possible (Onshape's native kernel).

## FRC examples (INFERENCE)

| Need | Approach |
|---|---|
| Polycarbonate intake funnel | Ruled surface from the floor edge at an angle, Thicken 0.093 in, or flat sheet metal if developable |
| Printed duct or shroud | Loft between sections with guides, Thicken 2 mm, Thickness analysis |
| Bumper corner cover | Sweep the profile along a frame-corner path; Face blend G1 |
| Handle grip | Revolve or loft, Fillet with G2, Zebra check |

## Do

- Keep the feature tree short and named (`SRF Funnel Side`, `THK Funnel 0.093`).
- Drive section sizes with Variables from the layout.
- Check continuity with analysis tools rather than by eye.
- Version before heavy direct edits on imported parts.

## Don't

- Don't use surfacing where a plate and bends would do.
- Don't loft through many profiles when a sweep plus guides does it.
- Don't Enclose with open (laminar) edges left.
- Don't trust smooth-looking shading; it hides G1/G2 breaks.
- Don't use community surface or "smooth" custom features.

## Common mistakes

- **Building from projected or trimmed curves,** giving wavy, heavy surfaces.
- **N-sided Fill used as the main surface** instead of as a patch.
- **Thicken failing** because the offset exceeds the minimum radius of curvature. Reduce thickness or smooth the curvature.
- **Forgetting surfaces have no mass:** the mass properties show zero.
- **Direct edits on an imported part with no version** to fall back to.

## Checks before done

- [ ] Only native features; feature list named.
- [ ] Combs smooth; zebra and connection analysis show the intended G-level at every key edge.
- [ ] Highlight boundary edges shows no open edges on the final solid.
- [ ] The solid has a material; mass is sensible; thickness ≥ the process minimum.
- [ ] Deviation from the reference or scan is within tolerance (if used).
- [ ] Export format matches the process (STL or 3MF, Parasolid or STEP, DXF for a developable cover).

## Sources

- Surface modeling overview (G0–G3, Thicken/Enclose, analysis, trimmed/projected-curve warning): https://cad.onshape.com/help/Content/PartStudio/surface_modeling.htm
- Surface features: https://cad.onshape.com/help/Content/PartStudio/boundary_surface.htm , https://cad.onshape.com/help/Content/PartStudio/fill.htm , https://cad.onshape.com/help/Content/PartStudio/loft.htm , https://cad.onshape.com/help/Content/PartStudio/sweep.htm , https://cad.onshape.com/help/Content/PartStudio/ruled_surface.htm , https://cad.onshape.com/help/Content/PartStudio/offset_surface.htm , https://cad.onshape.com/help/Content/PartStudio/constrained_surface.htm ,

https://cad.onshape.com/help/Content/PartStudio/bridging_curve.htm , https://cad.onshape.com/help/Content/PartStudio/face_blend.htm

- Solids from surfaces: https://cad.onshape.com/help/Content/PartStudio/thicken.htm , https://cad.onshape.com/help/Content/PartStudio/enclose.htm
- Direct editing: https://cad.onshape.com/help/Content/PartStudio/move_face.htm , https://cad.onshape.com/help/Content/PartStudio/replace_face.htm , https://cad.onshape.com/help/Content/PartStudio/delete_face.htm , https://cad.onshape.com/help/Content/PartStudio/split.htm
- Analysis: https://cad.onshape.com/help/Content/View/curve_surface_analysis.htm , https://cad.onshape.com/help/Content/View/connection_analysis.htm , https://cad.onshape.com/help/Content/View/zebra_stripes.htm , https://cad.onshape.com/help/Content/View/curvature_color_map.htm , https://cad.onshape.com/help/Content/View/reflection_analysis.htm , https://cad.onshape.com/help/Content/View/dihedral_analysis.htm , https://cad.onshape.com/help/Content/View/deviation_analysis.htm , https://cad.onshape.com/help/Content/View/draft_analysis.htm , https://cad.onshape.com/help/Content/View/thickness_analysis.htm , https://cad.onshape.com/help/Content/View/highlight_boundary_edges.htm
- FRC examples, print-wall minimum and polycarbonate thickness are INFERENCE/GUESS.

# onshape-top-down-master-layout

> **Summary (description):** Use when starting a robot, mechanism or product in Onshape whose parts share critical dimensions (frame, pivots, axles, envelopes), so that one master layout drives every Part Studio through Derived, in-context or multi-part design instead of copied numbers.

# Onshape top-down master layout

> **Native features only.** Never use FeatureScript: no custom features (including **Origin Cube**), Feature Studios, `/featurescript` or evalFeaturescript, and no "Any"-type variables. The layout is built from Variables, sketches, Planes and Mate connectors, the way a person would.

## When to use

- Kickoff of a new robot or a new mechanism with shared datums.
- Any time two or more parts must agree on a dimension: frame width, pivot height, shaft spacing, bumper envelope.
- Replacing a template that depended on the Origin Cube custom feature.

## Built-in replacement for Origin Cube

Origin Cube is a custom FeatureScript feature, so it is **forbidden**. Rebuild the same intent natively in the master layout Part Studio: 1. Named planes (`Plane` feature): `Frame Bottom`, `Frame Top`, `Bumper Top`, `Robot Centerline`, `Front Face`, `Rear Face`, offset from the default planes by variables. 2. A layout sketch on Top giving the frame perimeter and bumper envelope, constrained to the origin. 3. Explicit **Mate connectors** (Ctrl+m in the toolbar flyout; shortcut k toggles their visibility) on the origin and key points: - `MC Robot Origin` - `MC Drive Axle L/R` - `MC Arm Pivot` - `MC Elevator Base` 4. Version the result as the template (e.g. `T0 Template`) so new mechanism documents derive from an immutable source.

## Workflow

1. **Create the layout Part Studio** (`Master Layout`, or a top-level layout document for multi-document robots).
2. 6329 kept the complete assembly, reference sketches, a document directory and the tech binder in one top-level document, with subsystem documents linked through it.
3. **Variables first, at the top of the feature list:**
4. `#frame_l`, `#frame_w`, `#bumper_t`, `#ground_clear`, `#pivot_h`, `#max_ext`, `#max_h`.
5. Postman's drivetrain Part Studio puts its `#DTST`-style variables right after the master sketch and before the parts.
6. Use a **Variable Studio** for values shared across many Part Studios. Publish it only once the values are stable.
7. **Planes** offset from the defaults by variables, each named.
8. **Master sketches:**
9. `Top View Reference Sketch`: frame perimeter, bumper outline, wheel/module centers, robot centerline.
10. `Side View Reference Sketch`: ground line, frame rails, pivot points, arm or elevator travel arcs, height limit, extension limit.
11. Optional: a game-piece envelope and a field-element clearance sketch.
12. Every point that a part will need (pivots, axles, pulley centers) is a sketch point.
13. Measure from the origin, not from part faces.

14. **Mate connectors** on the key points, named for their mate role.
15. **Version the layout** (`L1 kickoff layout`).
16. **Derive into each mechanism Part Studio:**
17. Insert → Derived → pick the layout *version* → select only the needed sketches, planes and mate connectors.
18. Place with Base mate connector so the mechanism's origin sits at its pivot.
19. Postman's arm Part Studio starts with a folder `Master Sketches - Derived`, holding the sketches derived from the drivetrain Part Studio, before any part folders.
20. **Model parts with `Use` (u) on the derived geometry,** never by retyping numbers.
21. **Choose a method per part:** | Situation | Method | |---|---| | Shared datums, separate mechanisms | Derived from a versioned layout | | Tight intent known up front (gearbox, side plates + standoffs) | Multi-part Part Studio | | New part sized to existing parts in position | Create Part Studio in context | | Existing part that must follow assembly positions | Edit in context |
22. **Iterate:**
    - Change the layout, version it, then update the Derived references deliberately (a blue icon means out of date).
    - Review every downstream Part Studio after the update.

## What goes in an FRC layout (INFERENCE, from the samples and the 6329 thread)

| Item | Geometry | Driven by |
|---|---|---|
| Frame perimeter | Top sketch rectangle, centered on the origin | `#frame_l`, `#frame_w` |
| Bumper envelope | Offset of the perimeter | `#bumper_t` |
| Wheel / swerve module centers | Sketch points | Module offset variables |
| Ground and frame planes | Planes | `#ground_clear`, tube height |
| Pivots and axles | Side-sketch points plus mate connectors | `#pivot_h`, `#pivot_x` |
| Motion envelopes | Construction arcs and lines per pose | Arm length, elevator travel |
| Rule limits | Construction lines for max height and extension | Game manual values (cite the rule number in the sketch name) |
| Game piece | Simple circle or box at each handoff pose | Game manual dimensions |

- Keep one sketch per pose (`Pose Stow`, `Pose Intake`, `Pose Score High`) so reviewers can toggle them.
- Write the game-manual rule number into the sketch or variable name (e.g. `#max_ext_Rxxx`, where Rxxx is the actual rule number) so a rules update is easy to trace.

## In-context rules

- Rename each context immediately (e.g. `Ctx Arm at Stow`). The context snapshot appears under "Assembly contexts" and references show purple.
- Update contexts deliberately after the assembly changes. The primary instance has a solid arrow, secondaries a dashed one.
- You can't edit in context on an instance that references a revision.
- Avoid long chains of in-context references across many documents. Version the master and derive instead.

## Do

- Derive from a **version** of the master: it is faster, and you choose when to rebuild.
- Derive only mature, simple sources (sketches, a small Part Studio).

- Derive from the original source. Never chain A → B → C.
- Use one Derived feature per placement and assemble copies in the Assembly.
- Keep the layout light: sketches, planes and mate connectors, no solid detail.
- Record layout changes in the version description.
- Share reference sketches with partner teams through a versioned link, as 6329 did with a twin team.

## Don't

- Don't use Origin Cube, or any public custom feature, for datums.
- Don't derive from a large Part Studio that is still being edited.
- Don't use extra locations in one Derived feature to place many copies; that is slow, so assemble them instead.
- Don't type the same number into two parts; reference the layout.
- Don't reference another part's fillet or chamfer edges as the "layout".
- Don't link to live workspaces across documents for production work.

## Common mistakes

- **Deriving from the workspace "to stay current".** Every edit upstream regenerates downstream, and it breaks during experiments.
- **Layout dimensioned to tube bottoms or part faces.** A tube-size change moves everything. Measure from the origin.
- **No mate connectors in the layout.** The assembly then mates to part geometry that later moves.
- **Updating Derived references without reviewing.** Holes silently shift. Check each downstream Part Studio after an update.
- **Unnamed contexts** ("Context 1, 2, 3"). Nobody knows which assembly state they captured.

## Checks before done

- [ ] Variables and planes are at the top of the layout; all sketches are black and named.
- [ ] Mate connectors exist on every key point, and are named.
- [ ] The layout is versioned; every Derived feature points at a version, with no chains.
- [ ] No Origin Cube or other custom features anywhere (search the feature list).
- [ ] A test change to `#frame_w` (then undo) propagates correctly after updating references.
- [ ] Contexts are renamed and up to date.

## Sources

- Derived: https://cad.onshape.com/help/Content/PartStudio/derived.htm
- Updating references: https://cad.onshape.com/help/Content/Assembly/updating_references.htm
- In-context: https://cad.onshape.com/help/Content/Assembly/modeling_in_context.htm , https://cad.onshape.com/help/Content/Assembly/create_part_studio_in_context.htm
- Mate connector: https://cad.onshape.com/help/Content/PartStudio/mate_connector.htm ; Plane: https://cad.onshape.com/help/Content/PartStudio/plane.htm
- Variable / Variable Studio: https://cad.onshape.com/help/Content/PartStudio/variable.htm , https://cad.onshape.com/help/Content/VariableStudio/variable_studio.htm
- Onshape top-down blog: https://www.onshape.com/en/blog/top-down-design-onshape and the master model tech tip: https://www.onshape.com/en/resource-center/tech-tips/master-model-workflow . Both pages are JS-only, so the rules above come from the Help pages.
- Multi-part Part Studios: https://www.onshape.com/en/blog/onshape-best-practices-multi-part-part-studios
- 6329 structure (top-level document, reference sketches, origin): https://www.chiefdelphi.com/t/team-6329-the-bucks-wrath-2024-robot-reveal-dial/456061

- FRCDesign document setup: https://frcdesign.org/best-practices/document-setup/
- Observed practice: Onshape Explore "Postman" (master reference sketch, `#` variables, `Master Sketches - Derived` folder) and "Jackhammer" (Variables folder first, derived layout sketch).

# onshape-variables-and-configurations

> **Summary (description):** Use when an Onshape model needs variable-driven dimensions, a document-wide Variable Studio, or a configured family (lengths, sizes, options, alternate assemblies) with correct per-configuration part numbers and materials, or when deciding configuration versus branch.

# Onshape variables and configurations

> **Native features only.** Never use FeatureScript: no custom features (including "Parametric Part Numbers" style features), Feature Studios, `/featurescript` or evalFeaturescript, and **no "Any"-type variables** (they can carry FeatureScript values). Use the built-in Variable feature, Variable Studio, configurations and the Properties dialog.

## When to use

- Several features or parts share a value, e.g. wall thickness, tube size or a clearance.
- A part or assembly ships in several variants: tube lengths, 2- vs 3-stage elevator, wheel type, colors.
- Configured COTS parts: spacers, shafts, standoffs.
- Choosing between a configuration and a branch.

## Workflow – variables

1. **Insert the Variable feature at the top of the feature list,** above every feature that uses it. A Variable is a feature, so its position matters.
2. The Explore samples put a `Variables` folder first (Jackhammer) or right after the master sketch (Postman's `#DTST` variables).
3. **Name it:**
4. alphanumeric and `_`, case-sensitive, not starting with a digit, referenced as `#name`;
5. use a prefix per subsystem (`#dt_`, `#arm_`, `#el_`).
6. **Pick the type:** Length, Angle or Number only. Never Any.
7. **Use measured variables** (Distance, Length, Diameter) when a value must follow geometry. Use "From table" (CSV) only for simple lookups.
8. **Use expressions with units:**
9. `#wall * 2 + 0.5 in`
10. `#belt_teeth * 5 mm` (belt pitch length)
11. `#n_teeth * 0.25 in / pi` (#25 sprocket pitch-diameter approximation; use the exact `p/sin(180°/N)` from `engineering-math-proofs` for chain).
12. **For document-wide values, use a Variable Studio:**
13. Create it, add the variables, insert it into each Part Studio that needs it.
14. Keep "publish to workspace" unchecked while drafting, and publish once the values are stable.
15. An Assembly has its own Variable table for assembly-level values.
16. **Search the feature list with `:variable`** (case-sensitive) to audit where variables live.

## Workflow – configurations

1. **Open the Configuration panel** (Part Studio or Assembly). Configurations are created in a browser only.
2. **Add an input:** | Input | Use for | |---|---| | List | Discrete named options (`12 in`, `18 in`, `24 in`; `Flat Chisel`, `Point Chisel`) | | Checkbox | On/off options (`With Lightening`, `Lite`) | | Configuration variable | A continuous value within a range (`#tube_len` 1–48 in) |

3. **Rename the input tables meaningfully** (`Length`, `Wheel Type`, not `List 1`).
4. **Configure the features:**
5. Use "Configure features" mode (yellow banner) to change parameters per row, or right-click a feature → Configure suppression.
6. **Configure the properties per row:**
7. Part number, Name, Material, Appearance and custom properties, so each configuration has its own correct BOM line.
8. This replaces "Parametric Part Numbers" custom features: use **configured properties** (panel → Configured properties → Add property), or set them in the Properties dialog.
9. With a company numbering scheme, right-click the Part number column → Generate next part number.
10. **Use visibility conditions** to hide inputs that don't apply.
11. **Assembly configurations:**
12. Only mates, instances and patterns can be configured; mate connectors can't.
13. Postman's top assembly configures Wheel Type and color by suppressing the alternate instances.
14. Jackhammer's top assembly has a `Flat Chisel` configuration.
15. **Insert configured parts into assemblies from a version** of the Part Studio, and pick the configuration in the insert dialog.

## Worked example: a configured spacer family (native)

1. Variables at the top: `#od = 0.5 in`, `#id = 0.196 in` (#10 clearance; check against your hardware table).
2. List input `Length` with the stock lengths as rows (`0.125 in`, `0.25 in`, `0.5 in`, `1 in`). Each row sets `#len`.
3. A List rather than a continuous configuration variable, because every stocked length needs its own part number row. That choice is INFERENCE.
4. List input `Material`, with rows `Aluminum` and `Nylon`. Configure Material and Appearance per row.
5. Sketch: two concentric circles dimensioned `#od` and `#id`. Extrude it `#len`.
6. Configured properties → Add property → Part number. Type one number per Length × Material combination, e.g. `SP-AL-0250`.
7. If an input should not change the part number (e.g. a "compressed" pose of a spring), use the input's three-dot menu → **Exclude from properties**.
8. Parts that differ only in excluded inputs are treated as the same part by the BOM and release.
9. Version it as `Spacers v1`. Insert it from that version into assemblies, picking `Length` per instance.
10. Check: Mass properties change with `#len`, and the BOM shows distinct lines per length and material.

## Configuration or branch?

| Question | Configuration | Branch |
|---|---|---|
| Will every variant be built or stocked? | Yes | No |
| Will the variants live side by side in BOMs? | Yes | No |
| Is it an experiment you may throw away? | No | Yes |
| Does it change topology a lot (different mechanism)? | Usually no | Yes |
| Rule of thumb: **configuration = a family that ships; branch = an experiment.** | | |

## Performance

- The Performance panel flags configured workspace inserts whose regeneration exceeds 15 s (info) or 30 s (warning). Insert from versions to avoid it.
- Each extra configuration regenerates on demand, so keep heavy detail suppressible. A `Lite` checkbox suppressing fillets, threads and lightening cuts is the native replacement for "Simplify"; see `onshape-large-assembly-performance`.

## Do

- Put the variables above their first use and group them in a folder named `Variables`.
- Drive the configured dimensions with configuration variables, not dozens of list rows, when the value is continuous.
- Give every configuration row its own part number and name.
- Test every row after any edit: switch through them all and look for errors.
- Document the inputs in the feature or tab description.

## Don't

- Don't use Any-type variables or custom features to compute values.
- Don't configure mate connectors in an assembly; configure the mate or the instance.
- Don't use configurations for "what if" experiments; branch instead.
- Don't let all configurations share one part number.
- Don't insert configured parts from the live workspace into big assemblies.

## Common mistakes

- **A variable below the feature that uses it.** The feature errors or uses the old value. Drag the variable up.
- **Configured parts all show the same part number** in the BOM. Configure the property per row.
- **A suppressed feature in one row breaks downstream features** in that row. Check every row, not only the default.
- **Mixing units in configuration variables** (`12` vs `12 in`). Always give units.
- **Too many list rows** where a configuration variable would do, which bloats the table and slows regeneration.

## Checks before done

- [ ] Variables are at the top, named, and of type Length/Angle/Number only.
- [ ] Inputs are renamed; every row regenerates without errors.
- [ ] Part number, name and material are configured per row and appear correctly in the BOM.
- [ ] Configured parts are inserted from a version.
- [ ] No custom features or Any variables (search the feature list).
- [ ] Performance panel: no configuration warnings over 15/30 s.

## Sources

- Variable: https://cad.onshape.com/help/Content/PartStudio/variable.htm ; Variable table: https://cad.onshape.com/help/Content/PartStudio/variable_table.htm
- Variable Studio: https://cad.onshape.com/help/Content/VariableStudio/variable_studio.htm , https://cad.onshape.com/help/Content/VariableStudio/variable_studio_configurations.htm
- Assembly variable table: https://cad.onshape.com/help/Content/Assembly/assembly_variable_table.htm
- Configurations: https://cad.onshape.com/help/Content/PartStudio/part_studio_and_assembly_configurations.htm , https://cad.onshape.com/help/Content/PartStudio/managing_configurations.htm , https://cad.onshape.com/help/Content/PartStudio/configurations.htm
- Configured properties: https://cad.onshape.com/help/Content/PartStudio/configured_properties.htm ; visibility conditions: https://cad.onshape.com/help/Content/PartStudio/visibility_conditions.htm
- Properties: https://cad.onshape.com/help/Content/Home/properties_metadata.htm
- Performance panel thresholds: https://cad.onshape.com/help/Content/Home/performance_panel.htm
- Numeric fields: https://cad.onshape.com/help/Content/Home/numeric_fields.htm
- Versions/branches: https://cad.onshape.com/help/Content/Document/versions_and_history.htm

- Observed practice: Onshape Explore "Postman" (Wheel Type/color assembly configurations, `#` variables) and "Jackhammer" (Variables folder first, `Flat Chisel` configuration).

# 3D printing

# 3d-print-fits-and-tolerances

**Summary (description):** Use when designing FDM-printed parts that must fit, slide, snap, latch, or press together (sleds in a rack, lids, rails, dovetails, T-slots, crush-rib or heat-set-insert holes), when picking clearances or Bambu Studio compensation values, or when a printed fit must survive FRC robot vibration. Pair it with mech-design-fits-and-tolerances for machined or bought mating parts.

# 3D-print fits and tolerances (FDM, Bambu Studio)

Every claim is labelled **FACT** (a source in SOURCES.md, kept at /workspace/skill-drafts-print/3d-print-fits-and-tolerances/SOURCES.md), **INFERENCE** (reasoned from facts), or **GUESS**. Carry the labels into any design notes you write. For bearings, shafts, ISO fits, or metal parts, read **mech-design-fits-and-tolerances** first; this skill covers only the printed side.

## 1. What FDM accuracy really is

- A tuned Bambu-class printer usually holds about ±0.1–0.2 mm in XY on mid-size features. **INFERENCE**: Bambu publishes no accuracy figure for the H2S, so measure your own.
- Holes print small. Makelab says to add 0.1 mm to the radius of a critical hole (**FACT**, S2). Bambu lists shrinkage, elephant foot, wet filament, flow, and the seam as causes (**FACT**, S3).
- Z is set by layer height, so Z features land on layer multiples. Design Z gaps as whole layers, e.g. 0.2 or 0.4 at a 0.2 mm layer (**INFERENCE**).
- The first layer flares (elephant foot). Fix it with Elephant foot compensation or a 0.4–0.6 mm chamfer on bed edges that mate (**FACT**: the setting exists, S4. **INFERENCE**: the chamfer).
- Long thin walls bow and shrink more near the ends, so use the top of the clearance range on long slides (**FACT** for MJF/SLA, S2. **INFERENCE** for FDM).

## 2. Starting clearances (per side), then calibrate

| Fit | FDM per side | Use for |
|---|---|---|
| Press | 0.10–0.15 | crush-rib pockets, pins that stay put |
| Slip | 0.20–0.30 | alignment, snap-fit walls, sleds that should not rattle |
| Moving | 0.30–0.40 | hinges, sliders that run freely |
| Printed assembled | 0.40–0.50 | print-in-place |

All four rows are **FACT** (Makelab, S2). For a pin in a hole, the hole is pin + 2× the per-side value.

Smooth with no rattle is best done one of two ways (**INFERENCE**): - Slip-fit (0.20–0.25) on narrow contact ribs (0.5 mm proud), not on full faces. Ribs touch less area, so friction and print bulges matter less. - Moving-fit guidance plus crush ribs only at the end of travel. It runs freely and then locks solid where it stops.

## 3. Bambu Studio compensation

Find these under Process → Quality → Precision (**FACT**, S3, S4): - **X-Y hole compensation** enlarges closed holes only. The diameter grows by 2× the value. It does not work on layers where the hole is open at the side, and on those layers it can cause layer separation (**FACT**, S3). - **X-Y contour compensation** grows or shrinks the outer contour and leaves holes alone (**FACT**, S3). - **Elephant foot compensation** shrinks the first layer

(**FACT**, S4). - Order of operations (**INFERENCE**): 1. Dry the filament and tune flow/PA. 2. Print a coupon with compensation at 0. 3. Prefer changing CAD clearances over global compensation. Global values move every part, including press fits you already tuned. - Calibrate per filament and profile, then save a named "precision fit" process profile (**FACT**, S3).

## 4. Calibration coupon (always print first)

One small plate, about 10 g and 30 min, printed in the same filament, profile and orientation as the real part (**INFERENCE**). Include: - Slots over a 2.0 mm gauge fin that prints like your real wall, at 2.3/2.4/2.5/2.6/2.7 mm (0.15–0.35 per side). - Clearance holes 3.2/3.4/3.6 and heat-set insert holes 4.0/4.2/4.4 (CNC Kitchen M3: 4.0 nominal, about 4.2 printed is the tested sweet spot; **FACT**, S5, S6). - Pockets for any bought part you press in (e.g. Powerpole 16.0×8.3 and 16.2×8.5). - Then put the winning numbers into named CAD variables (CLR_SLIDE, CRUSH, INS). Never hand-edit individual features.

## 5. Crush ribs and detents

- **Crush ribs:** small vertical ribs 0.10–0.15 proud (the press-fit range, **FACT** S2), about 1 mm wide and 3–5 mm long, triangular or rounded. The rib crushes and the face does not, so you get zero-play location without needing exact holes (**INFERENCE**).
- Put them only where the part finishes its travel, or the whole stroke drags.
- Use 2–4 ribs per locating direction.
- Print them with layers running along the rib (vertical ribs on a vertical wall) so they deform without shearing off (**INFERENCE**).
- **Detents** (a bump into a dimple, about 0.3–0.6 mm) give a "click" and hold against drift, but a ramped detent can be cammed out. On a robot, use a detent only as a locator and pair it with a positive latch or screw (§8) (**INFERENCE**).

## 6. Cantilever snap/latch math

- Peak root strain: $\boldsymbol{\varepsilon = 1.5 \cdot h \cdot y / L^2}$. Minimum length: $\boldsymbol{L = \sqrt{(1.5 \cdot h \cdot y / \varepsilon)}}$, where h is the thickness in the bend direction, y the hook overlap and L the arm length (**FACT**, S7).
- Design strain for repeated use, from S7 (**FACT**):

| Material | Bends along layers (printed flat) | Bends across layers |
|---|---|---|
| PLA | ≈2% | ≈1% |
| PETG | 3–4% | ≈1.5–2% |

- PLA is stiff and brittle: keep it at ≤2% and use it only where the clip is seldom cycled. PETG forgives about 2× (**FACT**, S7). For robots, prefer PETG and design to the across-layer column anyway, because printed arms get abused (**INFERENCE**).
- Force: deflection force $F = 3 \cdot E \cdot I \cdot y / L^3$ with $I = b \cdot h^3/12$ (beam theory, **INFERENCE**). Assume PETG E ≈ 2 GPa (**INFERENCE**; check your filament's TDS).
- Hook angles (**FACT**, S7):
- Lead-in about 30°.
- Retaining face at 90° makes a joint that must be released by hand. About 45° makes one that pulls apart.
- Fillet the root at ≥0.5·h and taper the arm to about 0.5·h at the tip.
- Print orientation (**FACT**, S7): the arm should bend along its layers, so lay the bend plane parallel to the bed. If the part's shape forbids that, make the latch a separate part printed on its side and screw it on (**FACT**: S7 suggests a separate flat part).
- Add an over-press stop so a user can't push the arm past its strain limit (**INFERENCE**).
- The whitening test: if the root whitens, it has yielded. Lengthen the arm or reduce the overlap (**FACT**, S7).

## 7. Rails, dovetails, T-slots

- **Rails:** let the moving part's own walls run between fixed ribs. Keep the guided length over the width (L/W) ≥ about 1.5 so it doesn't rack and jam like a drawer (**INFERENCE**: anti-racking rule of thumb, no primary source). If the box is short, widen the contact spacing or add a centre rail.
- **Dovetails:** 45–60° flanks print without support when the tail points up (**INFERENCE**). Clearance goes on the flanks (slip 0.2–0.3), not the tip. Put a 0.3–0.5 chamfer on entry edges against elephant foot.
- **T-slots:** the head overhang prints as a bridge. Keep spans ≤ about 15 mm, or orient the slot vertically (**INFERENCE**). Use moving-fit clearance (0.3–0.4) because two surfaces stack their errors (**INFERENCE** from S2).
- On all of them, add a 1–2 mm lead-in chamfer or funnel at the mouth. Blind insertion in a tight robot space depends on it (**INFERENCE**).

## 8. FRC vibration and robustness

- Never rely on friction alone. Stack three things (**INFERENCE**):
- A positive latch with a vertical retaining face that has to be released by hand.
- Crush ribs at the end of travel, so the part can't rattle.
- A screw backup (one M3) for matches where you can't risk it.
- Thread into **heat-set inserts**, not plastic. Use CNC Kitchen hole sizes: 4.0 nominal and about 4.2 printed for M3, minimum wall 1.6 mm (**FACT**, S5, S6). Self-tapping into PETG strips after a few cycles (**INFERENCE**).
- **Creep:** a printed thermoplastic under constant load (a screw clamp, an interference fit) relaxes over time, worst when warm. PETG HDT is about 62–69 °C (Bambu PETG HF TDS, **FACT**, S8), and electronics bays get warm. So:
- Carry clamp loads through inserts and metal.
- Don't count on a plastic press fit to keep its preload over a season.
- Use thread locker on metal-to-metal joints only, because many lockers attack plastics (**INFERENCE**).
- Keep connector mating forces off the latch: give connectors a floating carrier (±0.5–1 mm) so misalignment doesn't load the printed parts (**INFERENCE**).

## 9. Print orientation checklist

- Sliding faces and rails go vertical (layers along the slide), so layer ridges run in the direction of travel (**INFERENCE**).
- Snap and latch arms bend along their layers (**FACT**, S7).
- Crush ribs go vertical. Insert holes print better vertical, with the axis along Z (**INFERENCE**).
- Large flat plates print flat. Put the mating face on the bed only if you've tuned elephant foot.

## 10. Worked example: a Pi 5 sled in a rack bay (PETG)

- Bay 72 wide, with 0.5 mm ribs on each side and 0.25 per side clearance (slip fit, S2), so the sled is $72 - 2\times(0.5+0.25) =$ **70.5 mm** wide.
- Guided length 115 mm over 70.5 wide gives L/W = **1.63**, which is ≥1.5 (OK).
- Latch: PETG beam h = 2.4, b = 8, L = 25 mm to the tooth, tooth y = 1.8 mm.
- $\varepsilon = 1.5\cdot 2.4\cdot 1.8/25^2 =$ **1.04%**, under the 1.5–2% across-layer PETG limit even though it's printed the favourable way.
- Over-press stop at 1.29%.
- Pad force about 3.6 N, which is easy to press with one finger.
- Retaining face is 90°, so sled load can't cam it open.
- Crush ribs 0.15 at the front (against the roof) and rear tongues 0.15 (in bracket notches): zero play at home, free running elsewhere.

- One M3 lock screw into a heat-set insert (4.2 hole) for matches.
- Coupon first, then set CLR_SLIDE, CRUSH and INS from it.

## Rules

- Coupon first, every new filament or profile.
- Put clearances in named CAD variables, labelled with their source.
- Never claim a fit "will work" without the coupon result. Say "INFERENCE: should slip-fit; confirm on coupon".

# 3d-print-strength-optimization

**Summary (description):** Use when designing, orienting, slicing, or checking a load-bearing FDM-printed part (FRC robot brackets, end plates, gussets, spacers, motor mounts) on the Bambu H2S or any FDM printer, when choosing a filament, when deciding printed vs aluminum, or when running FilaSim (filasim.com) to check a printed part or optimize its infill.

<!-- changelog (bridge retro, 2026-10-28): 1. New §7c: sliced-mass estimation for hard mass limits (thin members slice ~solid). 2. Calibration factors only transfer within the same geometry class. 3. Real slice in the target slicer is mandatory before calling a file ready. -->

# 3D-print strength optimization (FDM, FRC-focused)

> **MANDATORY FIRST STEP:** before choosing a concept or refining any design, run the `design-concept-gate` skill: list only the real constraints, challenge every self-imposed limit (height, part count, look, load case, print orientation), hand-calc 3-6 fundamentally different forms, and pick on numbers. No CAD or sizing until its decision table exists.

Labels: **FACT** (source in /workspace/skill-drafts-print/3d-print-strength-optimization/SOURCES.md, numbered [S#]), **INFERENCE** (reasoned from facts), **GUESS** (rule of thumb, verify). Related skills (go there, don't duplicate): whole-assembly load sharing → `load-path-simulation-for-assemblies`; Onshape Simulation setup → `onshape-simulation-fea-sanity`; consumables/spares/derates → `industrial-reliability-design-for-frc`; fasteners & preload → `mech-design-bolted-joints-and-fasteners`; release review → `frc-mechanism-design-review`; showing the math → `engineering-math-proofs`. Never use FeatureScript; Onshape GUI only.

## 0. The 60-second workflow

1. **Find the load path first** (`load-path-simulation-for-assemblies`, tool `/workspace/loadpath/`). Is the print in series (carries 100 %) or parallel with aluminum (carries ~EI or EA share, often <5 %)?
2. **Orient** so the main tensile/bending stress runs along the layers (XY), never pulling layers apart (Z).
3. **Size** with Z or XY strength from the table (§3), derated to ≤30 % of break (§9).
4. **Slice for strength**: more walls before more infill (§4), layer ≤ ½ nozzle, hot enough (§5).
5. **Check in FilaSim** (§8) using the "Layers" criterion; optionally export the infill-optimized 3MF.
6. **Print one, break one** if it's a structural part, then fill in the real break load.

## 1. Anisotropy, the one thing to remember

- FACT [S1]: CNC Kitchen tensile hooks, PLA: printed flat 63 MPa vs standing 31 MPa → layer bond ≈ 50 % of in-plane.
- FACT [S3–S9] Bambu TDS XY vs Z tensile: PLA 35/31, PETG HF 34/23, PETG-CF 35/29, ABS 33/28, ASA 37/31, PAHT-CF 92/47 MPa. Impact Z is 2–4× worse than XY (ASA 19.6 vs 4.9 kJ/m²). TDS specimens are 100 % infill and dried/annealed → best case; CNC Kitchen's real parts show a bigger gap than the TDS.
- INFERENCE: in Z, plan on **50 % of XY** unless you have your own test; the TDS ratio is optimistic.
- INFERENCE: stress concentrations at layer lines + notches make Z failures brittle and sudden, the failure mode FRC impacts find first.

## 2. Orientation

- FACT [S1]: inclining the layers helps: 30° → 44 MPa, 45° → 40 MPa (+27 % vs standing 31), 60° → 36 MPa, flat 63 MPa; the curve is S-shaped, so get at least 45° off the load if flat is impossible.
- Rules (INFERENCE from §1):
- Plates, gussets, end plates: print **flat** (load is in-plane = XY). Bearing bores then have layers across the hole axis, which is fine for radial load.
- L-brackets / motor mounts: the bend is the weak spot. Print so the bend's tensile fibres lie in XY (lay the "L" on its side), or split into two flat plates + bolts/aluminum angle.
- Standoffs / pins / hooks loaded in bending: never stand them up. Lay down or replace with metal.
- Bolt clamp load goes through Z (compression across layers); compression across layers is fine, tension isn't.
- H2S build volume 340×320×340 mm [S10].

# 3. Material table (FACT, Bambu TDS; xy / z)

| Material | E MPa | Tensile MPa | Bending MPa | HDT @1.8 MPa | Impact kJ/m² | FRC use (INFERENCE/GUESS) |
|---|---|---|---|---|---|---|
| PLA Basic [S3] | 2580 / 2060 | 35 / 31 (newer TDS 39/35) | 76 / 59 | 54 °C | 26.6 / 13.8 | prototypes, jigs, covers; creeps, brittle, softens in a hot car/pit |
| PETG HF [S4] | 1810 / 1540 | 34 / 23 | 64 / 48 | 62 °C | 31.5 / 10.6 | default robot plastic: tough, OK impact |
| PETG-CF [S5] | 2460 / 1340 | 35 / 29 | 70 / 48 | 68 °C | 15.7 / 10.7 | stiffer brackets, lower impact; hardened nozzle |
| ABS [S6] | 2200 / 1960 | 33 / 28 | 62 / 39 | 84 °C | 21.5 / 7.4 | needs enclosure/heated chamber |
| ASA [S7] | 2450 / 2120 | 37 / 31 | 65 / 40 | 92 °C | 19.6 / 4.9 | as ABS, UV stable |
| PAHT-CF (dry) [S8] | 3860 / 2180 | 92 / 47 | 125 / 61 | 170 °C | 22.8 / 13.3 | best structural FDM option; must be dry; hardened nozzle |
| TPU 95A HF [S9] | 9.8 / 7.4 | 27.3 / 22.3 | — | — | 123 / 86 | bumpers, compliant wheels/flaps, grip; not structure |
| Onyx (Markforged) [S11] | 2400 | 40 yield / 37 break | 71 flex | 145 °C @0.45 | — | reference only (search-result values; page blocked) |
| Al 6061-T6 [S12] | 68 900 | 276 yield | — | — | — | the comparison: ~20–40× stiffer than any of the above |
| INFERENCE: stiffness matters as much as strength. Aluminum is ~27× stiffer than PETG-CF, so a printed part in parallel with aluminum carries almost nothing, and a printed part alone deflects ~27× more. | | | | | | |
| PA/PA-CF absorbs moisture and gets softer/weaker when wet (FACT [S8] lists dry values separately; GUESS on magnitude). | | | | | | |

# 4. Walls vs infill vs pattern

- FACT [S2] (CNC Kitchen infill/shells): at 30 % infill, pattern changes strength only modestly (wiggle 45 kg, rectilinear 48, grid 52, honeycomb 57, triangular 60); between ~15–75 % patterns perform alike. **6 perimeters + 15 % infill ≈ 2 perimeters + 100 % infill.** Practical guidance: ≤4 perimeters and ≤50 % infill is usually the sweet spot for strength/weight.

- FACT [S1]: line width: adhesion peaks around 150 % of nozzle; hooks: 2 walls@100 % = 20 kg, 4@100 % = 33 kg, 3@133 % = 37 kg, 2@200 % = 39 kg.
- INFERENCE: bending and torsion stress live at the surface → walls + top/bottom shells do the work. For FRC plates ≤8 mm thick: 4–6 walls, 5–6 top/bottom layers, 25–40 % gyroid/grid usually ends up ~solid anyway. Around bolt holes, add walls (modifier or wall loops) so the clamp/bearing zone is solid.
- FACT [S13]: FilaSim's infill optimizer gave ~14–16 % higher stiffness at equal mass on its cantilever test; CNC Kitchen gradient infill ~30 % stiffer at equal weight for bending bars [S1].

## 5. Layer height, line width, temperature, cooling

- FACT [S1]: layer height ≤ ½ nozzle; 0.15 mm strongest in his test; 0.3 mm (on 0.4) ≈ half the layer strength.
- FACT [S1]: hotter = better bonding up to a point: PLA layer adhesion peaked 40 MPa at 230 °C (vs 60 MPa lying); PETG 32 MPa at 245 °C (vs 55). Too hot weakens again.
- FACT [S1]: heated chamber helps ABS Z: 21 MPa open vs 29 MPa at 65 °C chamber (75 % of flat 39). H2S has an active 65 °C chamber [S10].
- INFERENCE: reduce part-cooling fan on structural PETG/ABS/ASA/PA (keep enough for overhangs); a 0.6 mm nozzle + 0.3 mm layers gives thicker, better-bonded lines at speed and still meets the ≤½-nozzle rule.

## 6. Threads: inserts vs printed threads vs nuts (M3 PETG, FACT [S1])

| method | torque-out N·m | pull-out kg |
|---|---|---|
| heat-set insert | 3 | 119 |
| screw directly into plastic | 1 | 118 |
| Helicoil | 1 | 120 |
| captive nut from side | 2 | 86 |
| nut from bottom (pulls against plastic) | 2 | 166 |
| FRC rule (INFERENCE): **through-bolt with a nyloc nut + washer** wherever possible (plastic only in compression). Heat-set inserts when there's only one-side access. Never rely on printed threads for anything removed more than a few times. Washers spread clamp; plastic creeps → re-torque. | | |

## 7. Hole compensation and slicer profile (Bambu Studio)

- FACT [S14]: Process > Quality > Precision: **X-Y hole compensation** grows/shrinks closed holes only; **X-Y contour compensation** changes outer contours. Elephant-foot compensation is separate.
- GUESS starting points: model holes at nominal, print a test coupon (5 mm, 8 mm, 1.125" bearing bore, ½" hex) and set hole compensation from measured error (often +0.05 to +0.15 mm). Bearing bores: design for light press, verify on the coupon. Hex bores: add ~0.1–0.15 mm per flat (GUESS) and verify. Fit details → `3d-print-fits-and-tolerances` if saved.
- "FRC structural" process (setting names FACT [S14]; values INFERENCE/GUESS from §4–5): Strength > Walls > **Wall loops 4–6**; Top/bottom shells **5–6 layers**; Sparse infill **30–40 %** gyroid or grid; layer height **0.2 mm** on 0.4 nozzle (0.3 on 0.6); X-Y hole compensation per coupon; lower max fan for PETG/ASA/PA; a **modifier** (right-click part > Add modifier) to force 100 % infill / more walls around bolt holes and bearing bores; H2S chamber heating for ABS/ASA/PA (65 °C max [S10]).

## 7c. Sliced mass for hard mass limits (learned: est 27.1 g, real 31.09 g vs 30 g limit)

- Walls dominate thin parts: 3 walls x 0.42 mm ≈ 1.26 mm per side, so any member narrower than ~5 mm (in the sliced plane) or thinner than ~2x top/bottom shell thickness is effectively **solid**. Estimate member-by-member: solid if thin; otherwise perimeter shell + top/bottom shells + infill % of the core.
- A "calibration factor" (real/estimated) only transfers between parts of the same geometry class and same settings. Never apply a factor < 1 from a different part to get under a limit.
- Decide whether brim/skirt/supports count; read the model-only mass in the slice summary.
- Design to <= 90 % of the limit on the pessimistic estimate; then **real slice in Bambu Studio with the rule settings before saying "ready to print"**. If over: trim thick non-critical regions, then infill; keep walls on load paths.
- Log every real slice (part, settings, estimate, real) to improve the next estimate.

## 7b. Annealing

- FACT [S1]: PLA annealed 100 °C / 45 min: tensile 63.5 → 68.2 MPa (+7.5 %), layer adhesion unchanged (42 kg both), up to ~10 % dimensional change.
- INFERENCE: for FRC, annealing PLA is rarely worth it: small gain, warped holes, still layer-limited. Pick PETG/ASA/PAHT-CF instead. PA/PA-CF: dry before printing and store dry; follow the TDS drying spec.

## 8. FilaSim (filasim.com): what it is and how to use it

- FACT [S13][S15]: free, open-source (AGPL-3.0) browser FEA by Stefan Hermann (CNC Kitchen), listed as a tech demo. Rust→WebAssembly, runs **locally in the browser** (no upload). Import STL/3MF; voxel-style mesh. Boundary conditions: fixed, elastic support, slider; loads: force, pressure. Print settings: walls, line width, infill % + pattern (homogenized infill). Editable presets PLA/PETG/ABS/ASA (defaults e.g. PLA E 3500, σ 50, σz 35; PETG E 2100, σ 45, σz 34; interlayer shear default 0.6·σz). Failure criteria: **von Mises** and **"Layers"** (inter-layer tension σzz + interlayer shear). Extras: orientation sweep, modal, bed peel, infill optimization (max stiffness at fixed mass or same stiffness at less mass) exporting a **3MF with modifier volumes for Bambu Studio/OrcaSlicer**. Public GitHub repo returned 404 as of 2026-10-27.
- FACT limits: linear elastic, small deformation; no plasticity, contact, buckling, fatigue, creep, thermal. Use ≥4–6 cells through any bending thickness.
- INFERENCE workflow:
- Export the single part as STL/3MF from Onshape (right-click part > Export; GUI only) or from CadQuery.
- Set presets to **match §3 or your own test** (FilaSim defaults are higher than Bambu TDS, e.g. PETG σz 34 vs TDS 23; use the lower).
- Apply supports where bolts/aluminum actually hold it. If aluminum backs a face, use elastic or fixed support on that face; this is where the load-path step matters.
- Apply the part's **share** of load from `/workspace/loadpath`, not the total mechanism load.
- Run the orientation sweep → pick the printable orientation with the best "Layers" result.
- Read SF with the "Layers" criterion; require ≥3.33 vs break (≤30 %).
- Optional: infill optimize → export 3MF → open in Bambu Studio, check the modifiers kept.
- Alternatives (INFERENCE): Onshape Simulation (isotropic only; use Z strength as allowable; see `onshape-simulation-fea-sanity`); FreeCAD FEM/CalculiX (free, orthotropic possible with effort); hand calcs + `/workspace/loadpath`.

## 9. FRC: printed or aluminum?

- From `industrial-reliability-design-for-frc`: treat prints as consumables; load ≤30 % of measured break (GUESS-level rule). Carry spares in the pit.
- OK to print (INFERENCE): parts in parallel with aluminum (spacers, gusset fillers, sensor mounts, cable guides), end plates where a through-shaft on two bearings carries the bending, bumpers/covers/guards, compliant TPU parts, intake geometry meant to break before something expensive.

- Use aluminum (INFERENCE): the only path between a motor/gearbox and frame (series), cantilevered shaft supports, anything reacting a moment through a bearing bore, climber/endgame load paths (robot weight), drivetrain/swerve structure, parts above ~60 °C (PLA/PETG) or near motors, and anything whose failure disables the robot for the match.
- Inspect after every match: whitening (PETG), layer cracks, hole ovalization, creep at bolt heads.

## 10. Worked example (`/workspace/loadpath/examples/a_motor_mount.json`)

Inputs (GUESS scenario): PETG HF motor bracket, cantilever 60 mm, section 50 × 6 mm, 150 N tip load. 1. $I = b \cdot t^3/12 = 50 \cdot 6^3/12 = 900$ mm⁴; c = 3 mm; $M = 150 \cdot 60 = 9000$ N·mm → $\sigma = Mc/I =$ **30 MPa**. 2. Printed flat (XY, 34 MPa): SF = 1.13, **fails the 3.33 target**. Printed standing (Z, 23 MPa): SF 0.77 → breaks. 3. Fix A, thicken: t = 11 mm → I = 5546 mm⁴, $\sigma = 9000 \cdot 5.5/5546 = 8.9$ MPa → SF 3.8 ✓ (XY only). 4. Fix B, change the load path: bolt it to a 2x1x1/16 tube behind it → the tool shows the bracket carries 0.03 % (k 22.6 vs 74 093 N/mm), tube σ 2.95 MPa. The printed 6 mm plate is now fine, **if** the bolts are snug with no slop. 5. Choose B for the robot (stiffer, lighter); keep A only if there's no aluminum nearby. Verify with one break test.

# Mechanical design

# mech-design-bolted-joints-and-fasteners

**Summary (description):** Use when laying out how parts bolt together (gussets to tube, motor plates, bearing blocks, enclosure lids, desk brackets, 3D-printed mounts): choosing screw size and type, clearance vs tapped vs nut vs insert vs rivet, hole sizes, edge distance and spacing, locating features, screw length, and how to model it fast with the Hole feature and standard content.

# Bolted joints and fasteners: layout rules for real parts

> **Native features only, never FeatureScript.** Holes with the Hole feature, hardware from Onshape standard content. No fastener generators, no custom features, no Feature Studios.

Labels: FACT (sourced), INFERENCE, GUESS. Preload, torque, thread engagement (Le), tip-landing window and bearing stress formulas are in **engineering-math-proofs**; the VDI 2230 review order is in **frc-mechanism-design-review §4**. This skill is the geometry and the choice.

## 1. Pick the joint type (INFERENCE unless marked)

| Situation | Use |
|---|---|
| Thin sheet/plate to tube, FRC | Through-bolt + nylock nut, or rivet (fast, light) |
| Need to service often; thick aluminium (>= 2x D engagement available) | Tapped hole (engagement GUESS >= 1x D steel, >= 2x D aluminium) |
| Thin aluminium or polycarbonate that must take a screw | Rivnut / PEM press-in nut / insert |
| 3D-printed part | Heat-set insert or captive nut pocket; not printed threads (FACT Makelab: printed threads weak, min M6 FDM) |
| Wood (desk) | Threaded insert or cross dowel; wood screws only for light, rarely removed parts |
| Exact location between parts | **Dowel/locating pin + clearance bolts**; bolts in clearance holes don't locate |

## 2. Hole sizes (FACT: Noramark clearance chart; tap drills from MetalShopCalc/RivCut 75 % thread)

Metric clearance, ISO 273 fine / medium / coarse (mm): M3 3.2/3.4/3.6 · M4 4.3/4.5/4.8 · M5 5.3/5.5/5.8 · M6 6.4/6.6/7.0 · M8 8.4/9.0/10.0 · M10 10.5/11.0/12.0. Inch clearance, close / free: #4 0.116/0.1285 · #6 0.144/0.1495 · #8 0.1695/0.177 · #10 0.196/0.201 · 1/4 0.257/0.266 · 5/16 0.323/0.332 · 3/8 0.386/0.397. Tap drills (75 % thread): #4-40 #43 (0.089) · #6-32 #36 (0.1065) · #8-32 #29 (0.136) · #10-24 #25 (0.1495) ·

# 10-32 #21 (0.159) · 1/4-20 #7 (0.201) · M3 2.5 · M4 3.3 · M5 4.2 · M6 5.0 mm.

- Close: parts drilled together or screws that help locate. Free/medium: everyday default. Coarse: weldments, hand layout (FACT chart guidance).
- Onshape Hole > Clearance > Fastener fit offers Close, Free, Close (ASME), Normal (ASME), Loose (ASME) (FACT); use ASME B18.2.8 options when a print cites it.

## 3. Position tolerance from clearance (FACT: floating/fixed fastener formulas, Noramark)

- Floating fastener (bolt + nut, both parts clearance): **T = H - F** (diameter of position zone). 1/4 in screw in a 0.266 free hole → T = 0.016 in total.
- Fixed fastener (one part tapped): **T = (H - F) / 2** per part. #10 in a 0.201 free hole → 0.0055 in each.
- INFERENCE: a ±0.005 in laser (SendCutSend) uses most of a #10 fixed-fastener budget; use free fit or a floating (nut) joint on laser/router parts.

## 4. Edge distance and spacing (rules of thumb, GUESS unless code applies)

- Hole centre to free edge: **>= 1.5x D**, prefer **2x D** for tear-out (KingsForm hole guide). Structural steel: use the code value (AISC/Eurocode, ~1.2–1.25 d minimum) not this rule.
- Sheet/laser: hole edge >= 1–2x material thickness from part edge (SendCutSend 6061: 1.5–2x T; Machine Design DFMA).
- Pitch between bolts: GUESS >= 3x D so washers and tools fit.
- Minimum laser hole diameter >= material thickness (FACT SendCutSend).
- Tube wall: on 2x1x0.125 FRC tube, hole centres on the 1 in face sit at 0.5 in, i.e. 2.6x D for #10 (INFERENCE).

## 5. Screw length and stack (INFERENCE, uses the tip-landing window in engineering-math-proofs)

- Length = grip (all clamped parts + washers) + nut height + 1–2 pitches protrusion; round **up** to a stock length, then re-check it doesn't bottom in a blind hole.
- Standard content Auto-size rounds bolts **down** so they don't interfere, and nuts **up** from the bolt shaft (FACT): always re-check protrusion after Auto-size; it may be one size short for a nut.
- Keep the number of different screw sizes small (DFA, frc-dfm-manufacturing). FRC defaults: #10-32 and 1/4-20 (learnfrc units lesson).

## 6. Worked mini-example: 1/8 in 6061 gusset to 2x1x0.125 tube, #10-32

1. Joint type: through-bolt + nylock (tube wall too thin to tap: 0.125 in < 2x 0.19 in).
2. Holes: Onshape Hole, Simple, Clearance, Fastener fit Free → 0.201 in; select gusset **and** both tube walls in Merge scope so one feature cuts the full stack (FACT: multi-part scope).
3. Edge distance: centre 0.5 in from the tube edge (2.6x D OK); 0.40 in from the gusset edge (2.1x D OK).
4. Location: the gusset is located by the tube corner (or a 0.25 in dowel) — the bolts only clamp.
5. Grip = 0.125 + 1.0 (tube) + 0.125 + washer 0.05 ≈ 1.30 in; nylock #10 ≈ 0.23 in (check vendor) → 1.75 in screw.
6. Insert standard content: pre-select the gusset face so it fastens into ALL its holes (FACT), Auto-size, then add washers/nuts with Insert furthest (Shift+Insert) for the far side.
7. Load check: 150 lbf shear on one #10 in 0.125 in plate gives bearing 150/(0.19·0.125) = 6.3 ksi < 20 ksi (pass, method in engineering-math-proofs).

## 7. Gotchas

- Countersinks in thin sheet or polycarbonate pull through; use button heads + washers (frc-mechanism-design-review).

- Holes made with Extrude Remove get no Hole callout and standard content can't detect length or fine/coarse (FACT): always use the Hole feature for fastener holes.
- Powder coat shrinks holes (FACT chart): spec free fit or mask.
- Two parts each located by two tight bolts = over-constrained; one pin + one slot is the clean way (INFERENCE).
- Nylock nuts are one-use-ish for frequent service; use a thread locker spec instead and write it on the drawing.

## 8. Done when

- [ ] Every joint has type, size, hole fit, edge distance and length recorded.
- [ ] Locating is done by a feature or pin, not by bolt clearance.
- [ ] Standard content placed, Auto-size re-checked, hardware listed in the BOM.

Sources: /workspace/skill-drafts/SOURCES.md on the shared box.

# mech-design-fits-and-tolerances

**Summary (description):** Use when sizing any hole or shaft that has to fit something (bearing bores, pressed bushings, dowel pins, shaft-to-gear or pulley bores, hex bores, sliding parts, 3D-printed pegs and holes, laser-cut slots and tabs), when choosing an ISO 286 / ANSI fit, or when putting that fit into Onshape (sketch or Hole tolerance, Fit type, default tolerance block, drawing callout) so the part actually assembles.

# Fits and tolerances for real parts (and how to encode them in Onshape)

> **Native features only, never FeatureScript.** Tolerances go in sketch dimensions, Hole feature parameters and drawing callouts. No custom features, no Feature Studios.

Labels: FACT (sourced), INFERENCE, GUESS. The IT-grade table and the Ø12 H7/g6, k6, p6 worked numbers are in engineering-math-proofs; this skill picks the fit and the process offset.

## 1. Choose the fit by function (FACT: Noramark preferred-fits chart, ISO 286 hole basis)

| Function | Fit | Clearance at 10–18 mm (µm, + = clearance, - = interference) |
|---|---|---|
| Loose running, dirty, hot | H11/c11 | large |
| Free running (plain bearings, loose pivots) | H9/d9 | large |
| Close running (gearbox shafts in bushings) | H8/f7 | +16 to +61 |
| Sliding / locating that must move (guide pin) | H7/g6 | +6 to +35 |
| Locational clearance, snug, still assembles by hand | H7/h6 | 0 to +29 |
| Transition: gears/pulleys on keyed shafts, light-load inner rings (soft mallet) | H7/k6 | -12 to +17 (30 mm: -15/+19) |
| Tight transition (press) | H7/n6 | -23 to +6 |
| Locational interference: pressed bushings, not for torque by friction | H7/p6 | -29 to 0 |
| Medium drive (heavy press or heat) | H7/s6 | -39 to -10 |
| Force / shrink | H7/u6 | more |

Inch at 1/2 in (FACT, converted on the chart): H7/h6 +0.00114/0; H7/k6 +0.00067/-0.00047; H7/p6 0/-0.00114; H8/f7 +0.00240/+0.00063. ANSI B4.1 rough equivalents (FACT): RC1–2 ≈ g6, RC3–5 ≈ f7, LC from h6, LT ≈ k6/n6, LN ≈ p6, FN ≈ s6/u6. IT6/IT7 needs reaming, boring or grinding (FACT); a router, laser or printer will not hold it (INFERENCE).

## 2. Rolling bearing seats (INFERENCE: synthesis of SKF, NHBB, NTN fit pages; check the bearing catalogue)

- Rotating shaft, normal load, 17–100 mm: shaft **k5**; light load **j6**; heavy **m5**.
- Housing (stationary outer ring): **H7**; **J7** if load direction is indeterminate or there is shock.
- Rule: the ring that rotates relative to the load gets the interference; the other gets a slip/locational fit.

## 3. FRC / maker process reality (cite the source, these are not ISO grades)

- **1.125 in OD FRC hex bearings** (Chief Delphi 451819, community data = INFERENCE): VEX reamer gives ~1.126 in; 1678 roughs to 1.100 in then reams; as-machined press fit 1.122–1.124 in; one waterjet team uses 1.126 in for press, 1.127 in for normal. A typical FRC router can't hold a light fit consistently. Retain with Loctite 609/603, button heads overlapping a flanged bearing, or clamping blocks.
- Common reamed sizes: 0.375, 0.75, 1.125 in ±0.002 in (same thread).
- **Laser cut** (FACT, SendCutSend 6061 page): ±0.005 in; holes no smaller than material thickness; keep holes 1.5–2x thickness from edges. A ±0.005 process can't make a ±0.001 fit: cut undersize and ream.
- **Router**: inside corners >= tool radius + 0.005 in, test-fit a coupon (frc-dfm-manufacturing).
- **VEX hex bores** run ~0.003–0.005 in oversize (Chief Delphi 159650) — expect backlash; don't add more.
- **FDM 3D prints** (FACT, Makelab guide): holes print small, add 0.1–0.2 mm to the *radius*; minimum hole 2 mm. Per side: press 0.1–0.15 mm, slip 0.2–0.3 mm, moving 0.3–0.4 mm (pin-in-hole: hole = pin + 2x that). Heat-set insert hole = insert OD - 0.2 to 0.8 mm (another line on the same page says 0.1–0.2 mm: follow the insert maker's drawing). GUESS until you print a test coupon on your own printer and material.
- **Coatings** (powder coat, thick anodize) shrink holes and grow shafts (FACT, Noramark clearance chart note). Mask fits or size after coating.

## 4. Encode it in Onshape (FACT: help tolerance_options, hole, hole_thread_callout, default tolerances)

1. Model the **design size** (nominal for ISO fits, e.g. Ø12 H7; the chosen limit midpoint for FRC limit fits). Never bake a machine's cut offset into the design dimension; put it in a configuration (`Process = Router rough`) or the CAM/DXF step so the drawing stays true (INFERENCE).
2. Double-click the diameter (sketch or Hole) > tolerance icon > dropdown: Precision 0–6 places; Type = Symmetrical / Deviation / Limits / Min / Max / Basic, or for hole and sketch diameters **Fit / Fit with tolerance / Fit (tolerance only)** with Standard ANSI or ISO, Fit type (Clearance, Transition, Interference, User defined), Hole class, Shaft class.
3. Use **Limits** for FRC bores (e.g. 1.1235/1.1225) — shop-readable; use **Fit with tolerance** on ISO work (shows H7 plus the numbers).
4. Untoleranced features: set a default tolerance block (Onshape default tolerances library includes ISO 2768-f/m/c/v). ISO 2768-m is a sane general default for machined metal (INFERENCE).
5. Drawing: Hole/Thread callout pulls precision, tolerance and fit class from the Hole feature automatically. Sketch-circle bores need a normal diameter dimension with the tolerance.

## 5. Worked mini-example: bearing plate for a 1/2 in hex shaft

Part: 0.25 in 6061 plate, two 1.125 in OD flanged hex bearings, made on the team router. - Function: bearing must not walk out and must be replaceable → light press + flange retention. - Target bore: 1.123 in ±0.001 (Chief Delphi as-machined press 1.122–1.124). Router can't hold it → rough 1.100 in on the router, ream to final (1678 practice). Drawing: Limits 1.124/1.122, note "REAM". - Onshape: dimension the bore Ø1.123 with tolerance type Limits 1.124/1.122, and add a configuration `Router rough` that sets Ø1.100 for the CAM export only. - Check edge distance: bearing flange OD must clear the plate edge with >= 2x plate thickness of web (SendCutSend edge rule applied to a large hole, INFERENCE). - Verify: ream one coupon, press a bearing, record the measured bore in the team's process sheet.

## 6. Gotchas

- A "press fit" in printed PLA creeps and loosens; use heat-set inserts or a shoulder + screw (INFERENCE).

- Stack-ups: two clearance holes + one locating pin; never locate on two bolts (mech-design-bolted-joints-and-fasteners).
- Don't tolerance everything tight: tighter costs more time (learnfrc tolerances lesson). Only bores, pins and hex need it.
- Fits table numbers above are for 10–18 mm; look up the actual size step for other diameters.

## 7. Done when

- [ ] Every mating hole/shaft has a named fit and a process that can hold it.
- [ ] The tolerance is in the model (Hole/sketch tolerance) so the drawing callout picks it up.
- [ ] A coupon test is planned for any router/laser/print press fit.

Sources: /workspace/skill-drafts/SOURCES.md on the shared box.

# mech-design-shafts-bearings-gears-belts

**Summary (description):** Use when designing a gearbox, roller, intake, shooter, arm pivot, elevator drive, desk lift or any rotating axis: choosing shaft type and support, bearing type and retention, gear and belt/chain centre distances and adders, plate layout, and building it in Onshape with native sketches and variables instead of belt/gear generator FeatureScripts.

# Shafts, bearings, gears and belts (native Onshape workflow)

> **Native features only, never FeatureScript.** FRCDesign's belt lesson uses the Origin Cube and Belt & Chain Gen custom features: **do not**. Use the native sketch-and-variables method below.

Labels: FACT (sourced), INFERENCE, GUESS. Centre-distance and belt/chain length formulas and the 80T 18T/36T example (C ≈ 5.186 in) are in **engineering-math-proofs**; L10 bearing life too. Don't repeat them.

## 1. Shaft and support rules (INFERENCE unless marked)

- Support every loaded shaft with **two bearings** spaced as far apart as the package allows; a cantilevered shaft with one bearing wobbles and fatigues (frc-mechanism-design-review §3, §5).
- FRC: 1/2 in hex for drivetrains and heavy rollers, 3/8 in hex for light mechanisms (FACT learnfrc). Hex drives torque without keys; round shafts need a key (DIN shaft keys exist in standard content, FACT), set screw on a flat, or a clamp hub.
- Keep the gear/pulley close to a bearing; put the load between bearings, not outboard (INFERENCE).
- Axial retention: shaft collars, snap rings, or spacers stacked between bearings; draw the axial stack-up and leave 0.010–0.030 in end float (GUESS) so bearings aren't side-loaded.

## 2. Bearings (FACT unless marked)

- FRC: 1.125 in OD flanged hex bearings are standard (learnfrc; Chief Delphi 451819). Bearings for 3/8 in hex vary by vendor: read the OD off the vendor drawing (not verified here). Bore sizing and retention: **mech-design-fits-and-tolerances §3**.
- Industrial: seat the rotating ring with interference (shaft k5/j6/m5), the stationary ring with H7/J7 (INFERENCE from SKF/NHBB; verify in the catalogue).
- Flanges must bear on a face; put the flange on the side the load pushes toward or add a second retention (INFERENCE).
- Plastic plates: bearings walk in polycarbonate. Use aluminium bearing blocks or a bolted retainer (INFERENCE).

## 3. Gears (FACT: FRCDesign gears lesson; Chief Delphi 386702; Team 5026 wiki)

- Pitch diameter PD = N / DP; centre distance C = 0.5·PD1 + 0.5·PD2. FRC is mostly **20 DP**, 14.5° or 20°.
- **Adder** (extra centre distance for real-world tolerance): typical **+0.002 to +0.003 in**, always positive for gears (CD 386702); Team 5026 uses **+0.004 in** for 20 DP. GUESS: start at +0.003 in for router plates, check backlash on the first build.
- Thin gears (20 DP) want to be **sandwiched between two plates** (5026), not cantilevered.
- Motor plates: 1/4 in aluminium, 3/16 in minimum (5026).

- VEX hex bores run ~0.003–0.005 in oversize (CD 159650): expect some backlash from the bore, not only the mesh.

## 4. Belts and chain (FACT unless marked)

- FRC belts: HTD 5 mm pitch, 9 or 15 mm wide; belt length = teeth x 5 mm (80T = 400 mm) (FRCDesign belts).
- Belt adder: **none** for 5 mm belts (5026); a small **negative** adder only for low-torque, high-speed uses (CD 386702). engineering-math-proofs notes WCP-style 0 to -0.003 in for gearboxes (verify).
- Chain #25/#35: design **>= 1/4 in of tension adjustment** (5026) — slot one axle or add a tensioner.
- 3 mm GT2/HTD belts: sandwich between plates like 20 DP gears (5026).
- INFERENCE: a frame that bends changes centre distance under load and skips belts (frc-mechanism-design-review §5 cites this as the most common elevator failure). Stiffen, don't just add tension.

## 5. Native Onshape workflow (replaces the generator features)

1. **Variables** first (Variable feature or Variable Studio): `#DP = 20`, `#N1 = 14`, `#N2 = 60`, `#gearAdder = 0.003 in`, `#beltPitch = 5 mm`, `#beltTeeth = 80`.
2. **Layout sketch** on the plate plane: construction circles for each pitch diameter, diameters typed as expressions, e.g. `#N1 / #DP * 1 in` (FRCDesign: enter `(60/20)"` with Show Expression on).
3. Gears: dimension centre to centre = `(#N1 + #N2) / (2 * #DP) * 1 in + #gearAdder` (instead of a tangent constraint, so the adder is explicit). Belts: solve C by hand (engineering-math-proofs) and type it as a variable `#beltC`, or sketch the belt as two tangent lines + arcs and dimension the pitch-length expression if you need it to drive (INFERENCE).
4. Put a construction point at each axle centre; Hole/bore features in the plates `Use` these points (onshape-in-context-and-master-model, onshape-top-down-master-layout).
5. In the assembly, use **Gear relations** (and rack/screw relations) for motion; Simulation does not support relations, so replace them for FEA (onshape-simulation-fea-sanity).

## 6. Worked mini-example: one-stage 14T:60T, then 18T:36T belt

- Gear C = (14 + 60) / (2·20) = 1.850 in + 0.003 = **1.853 in** between motor and output axles.
- Belt C (80T HTD 5 mm on 18T/36T) = **5.186 in**, no adder (engineering-math-proofs; 5026).
- Plates: two 1/4 in 6061 plates, 1.125 in bearing bores at the output and roller axles, motor face holes on the motor plate. Gears sit between the plates. Spacers set plate gap = gear face + 2 x bearing flange + float.
- Onshape: one multi-part Part Studio (`Gearbox`), Variables → layout sketch → plate extrudes → Hole features for motor screws → bores with Limits tolerance → mirror the second plate. Assembly: Revolute mates + Gear relation.
- Check: overall ratio 60/14 x 36/18 = 8.57:1; confirm with engineering-math-proofs motor math.

## 7. Gotchas

- Typing PD without units in an inch document reads as inches; in mm documents add `* 1 in` (INFERENCE from numeric-field rules). Don't mix unitless and length values.
- The FRCDesign learning pages are written around FeatureScripts; take the math, skip the tools.
- A tangent-constrained pair of pitch circles has **zero** adder; real gears bind (CD 386702).

## 8. Done when

- [ ] Every shaft has two bearings, an axial stack-up and retention.
- [ ] Every centre distance is a named variable with its adder documented.

- [ ] Tension adjustment exists for chain; belts have a verified centre distance.

Sources: /workspace/skill-drafts/SOURCES.md on the shared box.

# FRC

# frc-competition-readiness

**Summary (description):** Use in the last two weeks before an FRC event and between matches to run the readiness gate: practice-vs-competition robot parity, spares and timed swap drills, battery and electrical checks, pit checklists, driver targets, and a go/no-go review.

# FRC competition readiness and pit operations

Labels: FACT (source), INFERENCE, GUESS. Source keys in `/workspace/frc-design-kb/top-teams.md`. `reliability.md` has a general maintenance list; this is the gated readiness process.

## 1. Two robots

- FACT [S14]: 2056 builds two complete robots yearly. FACT [S12]: 1690 practice robot weeks 3–4, competition robot week 6. FACT [S19]: 3847 practice machine and final machine. FACT [S21]: 1323 runs the competition robot as little as possible. FACT [S17]: 118 two identical driver stations.
- [ ] Every practice-robot change gets a change note and is mirrored before shipping.
- [ ] Parity: same firmware, code commit, configuration constants.

## 2. Swap modules and spares

- FACT [S7]: 1002 swaps the intake with 2 bolts; electronics pan comes out with only a drill. FACT [S13]: 2056 hinged ramp for battery access.
- [ ] Every fuse/wear part from the design review has qty_spare ≥1 (GUESS ≥2 if it broke in practice).
- [ ] Timed swap drills (GUESS gates): fuse/shear pin ≤2 min; intake or end effector ≤5 min; swerve module ≤10 min; motor ≤10 min. Record best and median. FACT [S8]: 1002 preseason skills drills.
- [ ] Spares kit labeled by subsystem and part number with a contents sheet.

## 3. Electrical and battery

- [ ] Every battery analyzed and labeled with internal resistance and date (FACT [S8]); retire GUESS >1.5× fleet median.
- [ ] Fleet GUESS ≥8 match-ready batteries for a 3-day event, with a rotation log.
- [ ] Pull-test every connector (FACT [S13] strain relief), tug main breaker lugs, wiggle while enabled.
- [ ] CAN: zero bus-off errors in a 10-minute run; star or segmented topology (FACT [S17]).
- [ ] Spare radio, controller and power distribution with a written swap procedure.

## 4. Software and driver gate (hardware owners sign off; the team writes robot code)

- [ ] Drivers ≥N practice hours (GUESS N=10 before event 1). FACT [S12]: 1690 schedules around driver time.
- [ ] Every auto 10/10 in a row on the practice field.
- [ ] Fallback auto and manual override for each automated function.
- [ ] Safe-state transitions verified (FACT [S4]: 1771).

## 5. Pit checklists

**Pre-match (≤5 min):** fresh battery latched and logged; bumpers correct color and tight; chain/belt tension, cracked plates, loose wires; mechanisms in start config, encoders zeroed or absolute; driver station connected, auto selected and read back aloud. **Post-match (≤10 min):** download logs, note current spikes and brownouts; inspect fuse parts and FMEA weak links; spot-check critical joint torque (FACT [S17] driver setting = screw size); log issues in the failure log and decide fix now / tonight / watch. **End of day:** re-tension rope/belt/chain (FACT [S23] 1678); full nut-and-bolt pass; charge batteries; restock spares.

# 6. Ship and inspection gate (T-7 days)

- [ ] Weigh the exact inspection configuration, margin ≥1 lb (GUESS).
- [ ] Frame perimeter and extension measured at every mechanism limit.
- [ ] Bumpers checked against this year's manual.
- [ ] Packing list: tools, spares, batteries, chargers, laptop with toolchain, printed checklists.

# 7. Go / no-go review

```
Event: ____  Date: ____  Robot: competition
FMEA High open: __ (must be 0)   Test pyramid all levels passed: Y/N
Swap drill medians: fuse __ / module __ / swerve __
Batteries ready: __   Driver hours: __   Auto 10/10: Y/N
Weight margin: __ lb  Parity with practice bot: Y/N
Decision: GO / GO-WITH-ACTIONS / NO-GO   Actions: …
```

# 8. During the event

- FACT [S17]: 118 "One Match at a Time": fix the biggest risk to the next match first.
- [ ] Scouting-informed strategy huddle before each match; cut features that aren't paying off (FACT [S14]).

# frc-defense-and-robustness

**Summary (description):** Use when an FRC robot must survive or play defense: bumper construction and rules, frame rail and corner strength, protecting mechanisms that leave the frame perimeter, pushing-force and traction math, tipping under side hits, and the pinning and contact fouls that shape a defensive design. Also use when reviewing a robot after it was damaged by contact. Mechanical only.

# FRC defense and robustness

Labels: FACT (sourced, keys in /workspace/skill-drafts-frc/SOURCES.md) / INFERENCE / GUESS. Rule numbers are from the 2026 manual [R1]. Re-check them every kickoff because the numbering moves. Math: `engineering-math-proofs` (Tipping and CG; Beams; Fasteners). Drive sizing: `frc-subsystem-drivetrain`. Onshape GUI only, never FeatureScript.

## 1. Requirement questions

- [ ] Will we play defense, or only survive it? (This sets drivetrain traction and ratio priorities.)
- [ ] Which mechanisms leave the frame perimeter, how far, and for how long each cycle?
- [ ] Which parts are fragile if hit at full speed: intakes, wheels, sensors, cameras?
- [ ] Where is the battery and CG, and how high? (That sets side-hit tipping.)
- [ ] What fouls apply this year to pinning and contact outside the perimeter (G418, G415 in 2026)?

## 2. Bumper rules that drive the design (2026 [R1]; verify each year)

- FACT R402: padding is 2.25 in of solid foam over a 4.5 in tall backing, with a cloth cover, attached rigidly and robustly.
- FACT R401: gaps under 1.25 in are allowed. One larger gap is allowed only if at least 5 in of perimeter on each side of each corner is protected.
- FACT R404: hard parts may extend at most 1.25 in beyond the frame perimeter, and the padding must stand ≥ 2.0 in beyond any hard parts.
- FACT R405: the bumper zone 2.5–5.75 in above the floor must be filled.
- FACT R408: 135 lb max including bumpers and battery.
- FACT [B1]: the KitBot bumper uses 3/4 in × 5 in plywood backing. The FIRST Bumper Guide allows other backing materials, including metal.

## 3. Architecture choices for robustness

| Choice | Option A | Option B | Notes |
|---|---|---|---|
| Bumper style | Continuous C-shaped or ring around corners | Separate side segments | FACT [B2]: experienced teams favor continuous wrap around corners. |
| Bumper mount | Brackets bolted to the frame, plus pins or latches | Velcro or straps only | INFERENCE: hits need a rigid mount (R402 "rigid, robust"). |
| Frame rail | 2×1×1/16 6061 with bumpers | 1/8 in front and back rails | FACT [D7]: 1/16 is common with bumpers. Some teams use 1/8 on the front and back. Without bumpers 1/16 would be destroyed. |
| Corners | Corner gusset + 2–4 in angle | Plain tube joint | FACT [B2]: angle brackets at the corners and bumper-to-frame gaps ≤ 1/4 in. |
| Out-of-perimeter | Retract inside on command | Stay deployed | INFERENCE: retract under contact, and design it to break away (a fuse) when it can't. |

| Choice | Option A | Option B | Notes |
|---|---|---|---|
| mechs | | | |
| Drivetrain for defense | Tank / 6WD | Swerve | FACT [D6]: not decisive. Tank resists lateral pushes better. Swerve can chase and keep position. |

# 4. Pushing and traction math

- Max push ≈ μ·W_on_wheels (FACT [D5] approach). AndyMark blue nitrile roughtop μ 1.19, pebbletop 1.24 [D5]. Colson about 1.0–1.2, wedgetop about 1.2–1.4 in older benchmarks [D6b].
- To actually reach μ·W, the gearing and current limit must deliver the matching wheel torque. Check this with the traction-current example in `frc-subsystem-drivetrain` §7.
- Lower gearing wins pushing matches at the cost of top speed (INFERENCE). A two-speed gearbox is weight and complexity.
- Weight on the wheels is the lever: run at the R408 limit and put ballast low (INFERENCE).

# 5. Tipping under a side hit

- Static check: a horizontal push F at bumper height h tips the robot about its far wheel line when F·h > W·d, d = horizontal CG distance to that line (eng-math Tipping and CG; INFERENCE simplification that ignores the dynamic impulse).
- Safety factor SF = W·d / (F·h). The highest F a robot can push with is about the opponent's μ·W_opp.
- A dynamic hit adds impulse. Apply ≥ 1.5× on SF for robots with tall extensions (GUESS; no measured FRC impact forces found).

# 6. Worked mini-example: 135 lb swerve robot hit by an equal robot

- Opponent push F = 1.2 × 135 = 162 lbf (μ 1.2, FACT-range [D5]).
- Bumper mid-height h = (2.5 + 5.75)/2 ≈ 4.1 in (FACT zone R405). CG to the far wheel line d = 12 in (GUESS; a 26 in frame with modules inset).
- SF = 135 × 12 / (162 × 4.1) = 1620 / 664 ≈ 2.4 → safe when stowed (INFERENCE).
- With the elevator up and the CG moved 4 in toward the hit side: d = 8 in → SF ≈ 1.6. Against the 1.5× dynamic gate it is marginal. Decision: retract before contact, or keep the extended CG ≤ 3 in off center (INFERENCE).
- The same math in Onshape: configuration `Extended`, Mass properties → CG location → measure to the wheel contact line.

# 7. Protecting mechanisms

- [ ] Everything fragile sits inside the frame perimeter or behind a bumper when stowed.
- [ ] Out-of-perimeter parts (intakes) have a sacrificial bumper or polycarbonate face, and a mechanical fuse (shear pin, breakaway pivot) sized below the structure's strength (see `frc-mechanism-design-review` fuses).
- [ ] Wheels and modules stay inside the bumper footprint so bumpers take the hit, not the module (INFERENCE).
- [ ] Cameras and sensors in recessed or guarded mounts; connectors latched (see `industrial-reliability-design-for-frc` §5).
- [ ] Battery strapped so it can't shift under a 162 lbf hit (INFERENCE from §6).
- [ ] Hard parts ≤ 1.25 in beyond the perimeter (R404) and nothing sharp that could catch an opponent's bumper.

## 8. Rules that shape defensive play (2026 [R1])

- FACT G418: no pinning longer than 3 s. The count ends at 72 in of separation or when the pinned robot moves away. Minor foul (5 pts) plus a major foul (15 pts) for each further 3 s.
- FACT G415: contact by a non-bumper part outside your frame perimeter, inside the opponent's frame-perimeter projection, is a minor foul. Mechanical consequence: retract extensions while defending (INFERENCE).
- Hand these to the drive coach and programmers as mechanical constraints. AI does not write strategy code.

## 9. Failure modes and fixes

| Failure | Evidence | Fix |
|---|---|---|
| Frame rail crushed or bent | FACT [D7] | 1/8 in front/back rails, or 1/16 with bumpers and internal bracing |
| Rivnut/thread pull-out in 1/16 wall | FACT [D7]: 10-32 gets only about 2 threads | Through-bolts with nuts, or rivnuts; don't crush the 2 in face |
| Corner damage, bumper shifts | FACT [B2] | Continuous bumper, corner angle 2–4 in, gap ≤ 1/4 in |
| Intake bent by a robot corner | FACT CD 468927 axle bent ~15° | Retract; reinforced polycarb; fuse |
| Swerve gears chewed under hard decel | FACT CD 472000 | Hand the accel-limit number to the programmers |
| Wires unplugged on hits | FACT CD 472000 | Latching connectors; strain relief |

## 10. Review checklist

- [ ] Bumpers checked against R401–R405 and R408 using this year's manual.
- [ ] Bumper mount rigid; corner brackets present; gap ≤ 1/4 in.
- [ ] Frame wall chosen with bumpers in mind; no threads cut directly in 1/16 wall on load paths.
- [ ] Push force μ·W computed, and the current limit can deliver it.
- [ ] Tipping SF computed stowed and extended; ≥ 1.5 extended (GUESS gate).
- [ ] Every out-of-perimeter part has a retract plan and a fuse.
- [ ] Pinning and contact rules written into the driver notes.

## Gotchas

- Bumper rules change most years: heights, cover colors, numbers. Never reuse last year's bumpers without re-checking.
- A robot that pushes well but browns out loses. Check the supply-current limits with the drivetrain skill.
- Defense exposes your own intake, so test hits on the practice bot before events.

## Hand-off to programmers (numbers only; AI writes no robot code)

- [ ] Drive current limits that deliver μ·W push without brownouts.
- [ ] Max accel and decel limits that protect the gears (CD 472000) and keep the tipping SF.
- [ ] Which mechanisms must auto-retract before contact, and their stowed positions.
- [ ] The G418 3 s pin count, so the driver dashboard can show a timer.
- [ ] Robot mass and CG height, stowed and extended, from Mass properties.
- [ ] Write these in the mechanism's spec doc, with units, and update them when hardware changes.

## Done when

Bumpers pass a self-inspection against the current manual, the tipping SF passes stowed and extended, every exposed mechanism has a retract or fuse plan, and a 10-hit test (GUESS gate) against a practice robot leaves no damage or loose parts.

# frc-dfm-manufacturing

> **Summary (description):** Use when turning reviewed FRC CAD into parts: buy vs make, material and process (router, laser, mill, lathe, 3D print, sheet bend), DFM/DFA part-count reduction, jigs and mistake-proofing, and a cut list and fabrication queue.

# FRC DFM, DFA and manufacturing flow

Labels: FACT (source), INFERENCE, GUESS. Source keys in `/workspace/frc-design-kb/top-teams.md`. Print orientation and packaging rules are in `industrial-design.md`.

## 1. Buy before make

- FACT [S14]: 2056: buy everything you can; make only what you can't and are capable of. FACT [S23]: 1678 reuses COTS end-effector parts across revisions.
- [ ] For each custom part, write why COTS fails (geometry, weight, cost). No reason → buy.
- [ ] Order long-lead items early (FACT [S14]). GUESS: anything with lead time >5 days ordered by day 3.

## 2. Part-count reduction (Boothroyd DFA)

FACT [I7]: part count is the biggest lever on assembly cost. A part stays separate only if it (1) moves relative to its neighbor, (2) must be a different material, or (3) must be separate for assembly or service. Otherwise merge it. - GUESS: DFA efficiency ≈ 3 s × N_min ÷ total assembly time; target ≥0.3 on subsystems. - FRC examples: tapped plates instead of nut+washer (FACT [S13] 2056); bumpers as structure (FACT [S20] 4414); swerve top plate as structure (FACT [S21] 1323).

## 3. Material and process

| Need | Default | Source |
|---|---|---|
| Bent sheet brackets | 5052-H32 | FACT [S23]: 1678, 6061 cracks when bent |
| Wear/sector gears | 7075 or steel | FACT [S20] 4414 |
| Flywheel mass | Steel rim on aluminum hub | FACT [S7] 1002 |
| Light stiff plates, fast iteration | Plywood, MAXComposite, polycarb | FACT [S3]: 1771 plywood frame stiffer and ~10 lb lighter; FACT [S19]: 3847 wood on practice, composite on final |
| Impact-tolerant plates | 1/4 in polycarbonate | FACT [S7] 1002 |
| Practice-robot prints | PETG (fails early, exposes weak links) | FACT [S21] 1323 |
| Competition prints | Onyx / PA-CF | FACT [S21][S13] |
| Light high-stage arms | Carbon-fiber tube | FACT [S15] 971 |

## 4. Process rules (GUESS until your shop measures)

- Router/laser: inside corners ≥ tool radius + 0.005 in; bearing holes cut undersize and test-fit on a coupon first; pocket webs ≥0.125 in.
- Box tube: standard hole pattern (e.g. 0.5 in grid) so a drill jig works (FACT [S8]: 1002 wood drill jigs).
- Sheet bend: inside radius ≥ thickness; 5052 not 6061 [S23].
- 3D print: load along layers; shear pin in hubs so the print isn't the fuse [S20].
- Lathe: standoffs and spacers from a standard-length table.

## 5. Mistake-proofing (poka-yoke, FACT [I15])

- [ ] Asymmetric hole patterns or keyed tabs so plates can't mount flipped.
- [ ] Every part engraved/labeled with part number and revision.
- [ ] Left/right mirror parts visibly different.
- [ ] Connectors keyed or color-coded per subsystem (FACT [S7] 1002).
- [ ] Jigs for any operation repeated ≥4 times (GUESS).

## 6. Fabrication queue

- Critical-path parts first (FACT [S12] 1690).
- Two of everything that gets swapped (FACT [S7][S14]).

```
part_no,rev,name,subsystem,material,stock,process,qty_robot,qty_spare,machine,est_min,status,owner,req_ids
INT-001,B,side plate,intake,PC 0.25,sheet,router,2,2,router,12,queued,,R-INT-1
```

- [ ] Every CAD part is in the cut list or COTS BOM.
- [ ] First-article check on critical dims, pass/fail with initials.
- [ ] After cutting: deburr, label, bag by subsystem, status `done`.

## 7. Revision control

- [ ] DXF/drawing names carry the rev letter (`INT-001_B.dxf`).
- [ ] Obsolete parts scrapped or red-X'd immediately.
- [ ] Post-CDR change note: `CN-## | part | rev A→B | reason | req IDs | approver`.

## Exit gate

Cut list 100% complete; long-lead items ordered; jigs ready; first articles pass; spares quantity set for every wear or fuse part.

# frc-mechanism-design-review

**Summary (description):** Use before releasing any FRC mechanism CAD for fabrication (CDR) to run a structured, numbers-first design review covering load cases, margins of safety, fatigue, bolted joints, stiffness/natural frequency, designed fuses, tolerance stacks, power transmission, sensors and serviceability.

# FRC mechanism design review (CDR checklist)

> **MANDATORY FIRST STEP:** before choosing a concept or refining any design, run the `design-concept-gate` skill: list only the real constraints, challenge every self-imposed limit (height, part count, look, load case, print orientation), hand-calc 3-6 fundamentally different forms, and pick on numbers. No CAD or sizing until its decision table exists.

Labels: FACT (source), INFERENCE, GUESS. Source keys in `/workspace/frc-design-kb/top-teams.md`. Tools in `/workspace/frc-design-kb/tools/`: `beam.py`, `motor.py`, `gearing.py`, `shooter.py`, `cycle.py`. Formulas live in `engineering-principles.md`; this skill is the review procedure.

## 0. Ground rules

- Reviewer is not the designer. Review against requirement IDs, not taste.
- Record findings as `F-## | H/M/L | location | finding | action | owner`.
- Outcome: PASS, PASS-WITH-ACTIONS, or FAIL. Fabricate only on pass, or pass-with-actions with no High open.
- FACT [S9]: a 5-minute reviewer hand calc caught a 1/2 in hex wrist at SF ≈ 0.7 (100 lbf hit on a 12 in lever). Always do the hand calc.

## 1. Load cases (write first)

| Case | Description | Load source |
|---|---|---|
| L1 | Normal operation | Motor torque at the current limit × ratio × efficiency |
| L2 | Hit/collision | GUESS 100 lbf at the extremity; replace with measured |
| L3 | Robot falls/tips onto mechanism | GUESS robot weight × 2 at the end |
| L4 | Handling (lift points, stepped on) | GUESS 50 lbf |
| L5 | Cyclic | L1 × cycles/match × matches; GUESS ~$10^4$ cycles/season |

Use the current limit (`motor.py`), not free speed. Gear efficiency ~90–95% per stage (FACT [S16]).

## 2. Margin of safety (NASA, adapted)

- MS = allowable / (FF × FS × applied) − 1 ≥ 0. FACT [I4].
- Fitting factor FF ≥ 1.15 at joints/fittings. FACT [I4].
- NASA metallic minimums: ultimate 1.4; yield 1.0 tested prototype, 1.25 protoflight. FACT [I3]. Bigger factors never replace testing.
- FRC targets (GUESS, loads poorly known): L1 FS ≥ 2 on yield; L2/L3 ≥ 1.25 or a designed fuse fails first (§6); L5 fatigue per §3; printed/composite ≥ 2 at discontinuities (FACT [I3]: NASA 2.0 for composite discontinuities).

## 3. Fatigue (arms, wrists, flywheels, hex shafts)

- Steel endurance ≈ 0.5 × UTS (cap ~290 MPa); aluminum ≈ 0.4 × UTS at $10^7$–$10^8$ cycles, cap ~19 ksi. Smooth-specimen values; notched parts much lower. FACT [I11].
- Apply Marin factors (surface, size) and Kf at holes, keyways, hex corners [I11].
- Goodman: σa/Se + σm/Sut ≤ 1/n [I12].
- [ ] Every cycled part has a fatigue check, or is on the spares list with a replacement interval.

## 4. Bolted joints (VDI 2230 / NASA-STD-5020 lite)

- FACT [I4]: torque-controlled preload scatters ~±25% lubricated, ±35% dry. Design so minimum preload still clamps after ~5% relaxation.
- FACT [I17]: VDI 2230 order: scatter → min clamp → embedding → min/max preload → fatigue → bearing pressure → thread engagement → slip.
- [ ] Thread engagement GUESS ≥1×D steel, ≥2×D aluminum, else rivnut/insert (FACT [I4]: insert allowable = lower of thread and pull-out strength).
- [ ] Locking feature chosen on purpose (FACT [I4]: NASA selection flowchart).
- [ ] Torque spec written. FACT [S17]: 118 sets driver torque = screw size (4 for 4-40, 10 for 10-32). Tools used within 20–90% of scale [I4].
- [ ] Polycarbonate gets fender washers [S11]; no countersinks in thin polycarb (GUESS).

## 5. Stiffness, natural frequency, deflection

- FACT [S15]: 971 stiffened an arm to ~10 Hz. FACT [S16]: frame vibration increased shot scatter.
- FACT (KB survey): the most-cited elevator failure is chain/belt skip from a bending crossbar or bowing frame. Check center-distance change under load.
- [ ] End-effector deflection under L1 ≤ 25% of placement tolerance (GUESS), via `beam.py`.
- [ ] fn = (1/2π)√(k/m_eff) ≥ 3× control bandwidth (GUESS).
- [ ] Long hex/thin plates in bending get box beams, tube plugs or extra bearings [S9].
- [ ] High-RPM rollers/flywheels checked for centrifugal growth of coverings.

## 6. Designed fuses

- FACT [S20]: 4414 shear pins in printed intake hubs. FACT [S15]: 971 series-elastic intake compliance.
- [ ] For every L2/L3 path, name the part that fails first: cheap, swappable ≤5 min (GUESS), in the spares kit.

## 7. Tolerance stacks

- Worst case for must-assemble interfaces (bearings, gear center distance, belt/chain C-C, swap modules); RSS for the rest. FACT [I9].
- [ ] Gear center distance GUESS ±0.003 in; belts use the CTC function plus a tensioner.
- [ ] A slot, tensioner or shim absorbs the stack (FACT [S11]: 254 shims gear-to-hex); datums marked on critical plates [I10].

## 8. Power transmission and sensors

- [ ] Ratio checked against the time requirement with `cycle.py` at the current limit and 10 V (GUESS brownout margin).
- [ ] Mechanical hardstops with soft limits inside.
- [ ] Absolute encoder or zeroing on every joint [S9]; beam-break or current detection for game pieces [S13].
- [ ] Wear parts flagged (7075 sector gears [S20]; rope/belt re-tension schedule [S23]).

## 9. Serviceability and hand-off

- [ ] Tool and wrench clearance for every pit-reachable fastener.
- [ ] Hand off to frc-dfm-manufacturing and frc-weight-and-cg-management.

## Review sheet

```
Mechanism: ____  Req IDs: ____  Reviewer: ____  Date: ____
Load cases L1–L5 written?  Y/N
Min MS (case, part): ____   Fatigue-checked parts: ____
Bolted joints checked: __/__  Fuse part: ____
fn est: __ Hz  Deflection at EE: __ in (allow __)
Worst-case stacks: ____   Sensors/zeroing: ____
Findings: F-01 … | Decision: PASS / PASS-WITH-ACTIONS / FAIL
```

# frc-prototyping-and-iteration

> **Summary (description):** Use in the first FRC build weeks, and whenever a mechanism fails its requirement, to run fast measured prototypes, steal-and-adapt proven designs, build a throwaway alpha robot, and iterate mechanisms through numbered versions with test data deciding each change.

# FRC prototyping and iteration

Labels: FACT (source), INFERENCE, GUESS. Source keys in `/workspace/frc-design-kb/top-teams.md`. `design-process.md` has the 72-hour kickoff plan; this covers how to prototype and iterate.

## 1. Steal and rate first

- FACT [S23]: 1678: "Steal from the best, invent the rest." FACT [S18]: 6328 prototyped 3 hopper concepts from 2017 robots (118, 1986, 125). FACT [S9]: 1002 redesigned from 2910, 125, 1678 ideas. FACT [S4]: 1771's elevator was inspired by 4414's.
- [ ] For each function list ≥3 prior-art designs with links (Chief Delphi, binders, 3847's FRC CAD Collection [S19]).
- [ ] Rate TRL (frc-systems-engineering). Prototype only what is TRL ≤4 for your team.

## 2. Prototype rules

- One question per prototype, written first (e.g. "Does a 2 in compliant wheel at 3 in compression grab the piece at ≥2 ft/s?").
- Real game piece, real speeds, real ratio. Cardboard, wood, polycarb, prints are fine.
- Adjustable: slots for compression, center distance, angle.
- Measure: FACT [S16] 971 filmed and analyzed prototypes (8 funnel concepts). Record n, successes, failure mode.
- Stress it: FACT [S16] 971 fed 5–6 balls into a 3-ball funnel over hundreds of trials.
- Time box GUESS ≤1 day per question, 3-variant sweep.
- FACT [S12]: 1690 spends days 2–5 only on prototypes; CAD starts day 6.

```
P-## | question | variant | settings | n | success | failure modes | video | decision
```

Decision rule (GUESS): adopt at ≥18/20 at game speed with the main failure mode understood.

## 3. Alpha robot

- FACT [S11]: 254 built a 2025 AlphaBot in 1 week. FACT [S18]: 6328's Alpha Bot was ~90% mechanically done by day 2, built to move fast and break things. FACT [S17]: 118 practice drivebases since 2009.
- [ ] Existing drivebase plus rough mechanisms; nobody polishes it.
- [ ] Output: lessons that change requirements or architecture, delivered by PDR (GUESS day 7–10).
- [ ] Software gets it immediately.

## 4. Numbered iteration

- FACT [S20]: 4414 rotor V1–V10, shooter V1–V11. FACT [S12]: 1690 intake ~10 iterations. FACT [S7]: 1002 swapped torn star wheels for round, dual shooter to drum.

- [ ] Each version: one number, one change hypothesis, test data. Two changes at once = ambiguous data. | Ver | Change | Why (data) | Result | Keep? | |---|---|---|---|---| | V3 | +0.25 in compression | V2 slipped 4/20 | 19/20 | yes |
- [ ] Iterate after events from the failure log (frc-reliability-testing). FACT [S14]: finish early and iterate.

## 5. CAD fidelity

- FACT [S20]: 4414 medium-fidelity block CAD: power transmission and critical geometry exact, plates simple until late.
- [ ] Game-piece path, frame perimeter, extension limits live in one master layout sketch.

## 6. Freeze

- [ ] Requirement met with margin (GUESS 20% on time gates).
- [ ] Remaining failure mode known, with fuse or spare.
- [ ] Mass within budget.
- After freeze, changes need a change note and mini-CDR. Do less, better [S14].

## Anti-patterns

Wrong game piece or speed; untracked tweaks; polishing the alpha; inventing when a proven open design exists; skipping prototypes without team heritage.

# frc-reliability-testing

**Summary (description):** Use when planning or running FRC robot reliability work: AIAG-VDA-style FMEA with Action Priority, a test pyramid from coupon to full match, abuse tests, qualification gates before events, and a failure log that feeds design changes.

# FRC reliability and test engineering

Labels: FACT (source), INFERENCE, GUESS. Source keys in `/workspace/frc-design-kb/top-teams.md`. `reliability.md` covers RPN FMEA, the FRC failure-mode list and wiring; this adds Action Priority, the test pyramid and qualification gates.

## 1. FMEA with Action Priority (AIAG-VDA 2019 style)

FACT [I6]: Action Priority (AP) replaces RPN; it looks up Severity, Occurrence, Detection with severity weighted first and returns High/Medium/Low. High = act or document why not. Simplified FRC table (GUESS; official table is paywalled): | S | O | AP | |---|---|---| | 9–10 (safety, disable, lost match) | ≥4 | High | | 9–10 | 2–3 | High if D ≥5, else Medium | | 7–8 (lose a scoring function) | ≥6 | High | | 7–8 | 4–5 | Medium | | 4–6 (degraded) | ≥6 | Medium | | else | | Low |

Row: `ID | item/function | failure mode | effect | S | cause | O | prevention | detection | D | AP | action | owner | due | new S/O/D` - [ ] Every Need has FMEA rows. - [ ] Every High has an action or signed justification at CDR. - [ ] Prefer lowering O (redesign, fuse) over lowering D (inspection). Severity changes only through design.

## 2. Test pyramid (FACT [I14]: many cheap low-level tests, few expensive end-to-end)

| Level | Example | Count (GUESS) | Gate |
|---|---|---|---|
| Coupon | Plate bend test (FACT [S19] 3847); print orientation | 3 per material | Before CDR |
| Component | Load test (FACT [S15] 971 lift at 38 lb); joint torque | Each new load path | Before assembly |
| Subsystem bench | Intake 20/20; elevator full travel ×20 | ≥20 consecutive passes | Before integration |
| Subsystem abuse | 971 fed 5–6 balls into a 3-ball funnel, hundreds of trials (FACT [S16]); L2 hit | ≥50 trials | Before event |
| Integrated | Full auto ×10, practice-field cycles (FACT [S19] 3847 built a field in week 1) | 10 autos, 2 h driving | Before event |
| Match-like | Defended scrimmage, battery swap, field reset | ≥10 matches | Before event |

- [ ] Every requirement ID maps to ≥1 test row [I13].
- [ ] Log: `date | test | config/rev | n | passes | fail modes | video`.
- [ ] Measure with video (FACT [S16]) and logs (current, position).

## 3. Deliberate weak-link discovery

- FACT [S21]: 1323 prints practice-robot parts in PETG to fail early and runs the competition robot as little as possible.
- FACT [S7]: 1002 assumes the intake will break and designs a 2-bolt swap.
- [ ] After every break pick one: redesign, fuse, or spare with an inspection interval. Never fix-and-forget.

## 4. Qualification gates

- FACT [I3]: NASA qualifies prototypes at 1.4× limit load, protoflight at 1.2×.
- GUESS FRC version: proof-load each structural subsystem once at 1.2× L1 on the practice robot; ≥30 min continuous-cycle burn-in on the competition robot before event 1.
- [ ] Only tested state transitions allowed (FACT [S4]: 1771 safe-state graph).
- [ ] Detection features: beam-break auto-retract; current-based jam detect with auto-reverse (FACT [S13] 2056); current-based climb detection (FACT [S22] 2910).

## 5. Failure reporting (FRACAS-lite)

```
FR-### | date/event/match | robot (P/C) | subsystem | symptom | root cause (5 whys) | fix type
(redesign/fuse/spare/procedure) | verified by test # | closed Y/N
```

- [ ] Review open FRs weekly (FACT [S19]: 3847 Tuesday Design Recap).
- [ ] Track matches between failures per subsystem; GUESS target ≥20 for any Need subsystem by district championship.

## 6. Electrical gates

- FACT [S13] 2056 strain-reliefs every connection; CAN joints soldered, ring or spring terminals. FACT [S17] 118 star CAN topology. FACT [S19] 3847 auto-reset breakers for radio and controller. FACT [S8] 1002 battery analyzer.
- [ ] Wiggle-test every connector while enabled: zero drops.
- [ ] Every battery analyzed and labeled with internal resistance; retire GUESS >1.5× fleet median.

## Exit gate (before first event)

No open High AP; every pyramid level passed for every Need; burn-in done; spares kit matches fuse/wear list; FR log reviewed.

# frc-robot-cad-organization

**Summary (description):** Use when setting up, restructuring or onboarding people into an FRC/FTC season's Onshape CAD — folder and document structure, concept/layout → subsystem → main-robot linking, part and assembly numbering, a native start template, COTS libraries, weight budget, design review and training path.

# FRC robot CAD organization in Onshape

> **Native features only.** Never use FeatureScript: no custom features (Origin Cube, Belt/Chain Generator, Laser Joint, Lighten, Simplify, part-number generators), Feature Studios, `/featurescript` or evalFeaturescript. The onboarding blog's "Custom Features" section is deliberately skipped. Everything below uses built-in tools as a person would. Never edit documents owned by another team or the team's admins without being asked; read and copy ideas instead.

## Target structure

```
<Season> (folder, owned by the team classroom/company; shared with Link permission)
├── 00 Templates       T0 Template (Part Studio + Assembly), drawing template, Lite/proxy parts
├── 01 Concept         layout sketch(es), crayon/KrayonCAD robot, Variables, field elements
├── A100 Drivetrain    Part Studios PS100…, Assembly A100
├── A200 Arm           PS200…, A200; A201 Intake, …
├── A300 …             one document per subsystem
├── A400 Electronics   electrical board, wiring drawing
├── 90 Robot           main robot assembly (imports subsystem top assemblies by version),
│                      document directory, tech binder, drawings index
└── 99 Archive         prototypes, abandoned concepts (versioned, read-only)
```

- This matches FRCDesign's Concept → Subsystem → Main Robot structure. The layout comes into subsystems by **Derive**; subsystem assemblies come into the main robot by **Insert** (import). Concept and Main are sometimes combined.
- 6329's public robot: a **top-level document** holding the complete assembly, reference sketches, a **document directory** and the **tech binder**, with subsystem documents reached through the directory. Measurements are made **to the origin**, not to the drive tubes. Reference sketches were shared with a twin team.

## Workflow

1. **Ownership and folders.**
2. The team classroom or company owns every document.
3. Make a season folder with subfolders per project; share with the classroom including Link permission.
4. Keep root-folder permissions minimal.
5. **Numbering (from the Postman sample):**
6. Part Studios `PS100`, assemblies `A100`, with the hundreds digit = subsystem: A100 Drive, A200 Arm, A201 Intake, A300 Spinner, A400 Electronics.
7. Put the team or robot prefix in names, e.g. `(TEAM-RB-PS100) Drivetrain`.
8. Part numbers: `<team>-<yy>-<subsys><nn>` (e.g. `1234-26-1003`, INFERENCE). Set them as part properties or **configured properties**, never with a part-number custom feature.
9. **Build the native T0 Template** (replacing Origin Cube), and version it `T0 Template`:
10. **Part Studio:** Variables first (`#frame_x`, `#frame_y`, `#tube_h`, `#bumper_t`, `#max_ext`). A layout sketch on Top with the frame perimeter and centerlines. Named planes (`Frame Top`, `Bellypan`, `Bumper Top`). A "Master Sketches –

Derived" folder ready to fill.
11. **Assembly:** origin instance or fixed layout part, explicit **mate connectors** named for each axis or location (`MC Pivot Arm`, `MC Intake Axle`), instance folders (`Hardware`, `Motors`, `Bearings`, `Electronics`, `Patterns`).
12. New subsystem documents start from **Copy** of the template document.
13. **Concept/layout first** (see `onshape-top-down-master-layout`):
14. Layout sketches for side and top views (as on Postman's "Top View Drivetrain Reference Sketch"): frame perimeter, bumper envelope, extension limits, pivot points, game-piece path.
15. Variables like `#DTST`-style drivetrain values in a Variables folder first.
16. Version it, then derive it into each subsystem at that version.
17. **Subsystem documents:**
18. One Part Studio per part family; folders per plate group; build one side and Mirror.
19. Hole features named by spec; sheet metal for bent parts; COTS from libraries.
20. The subsystem assembly is **fully constrained (rigid)** except intended motion.
21. **Instance folders by function** (Jackhammer: DRIVETRAIN, GEARS, HARDWARE, POWER, PATTERNS, BEARINGS; Postman: "#8-32 Nuts", "Battery/Radio").
22. Named mates (`Drive`, `Arm Pivot`); Replicate for repeated hardware.
23. **Main robot document:**
24. Insert each subsystem top assembly **from a version**. Few instances and few mates: Postman's top assembly has 5 instances and 4 Fastened mates.
25. Configurations (e.g. wheel type, color) suppress alternates.
26. A document directory (table or notes with links), the tech binder, a Render Studio tab and an assembly manual tab.
27. **COTS and hardware:**
28. Vendor parts from **FRCDesignLib** through the FRCDesignApp (the onboarding blog names MKCAD, but MKCad is deprecated and unmaintained per FRCDesign). Use KrayonCAD simplified assemblies for concept work and **Standard content** for fasteners. Insert parts only; skip any custom features the app offers.
29. Use proxies for heavy parts (see `onshape-large-assembly-performance`).
30. Never re-model COTS by hand unless no model exists.
31. **Weight and CG budget:**
32. A material on every part; mass overrides on COTS proxies.
33. A budget table per subsystem (e.g. Drive 45 %, Arm 20 %, Intake 12 %, Electronics 15 %, Reserve 8 %, INFERENCE).
34. Check mass properties in each subsystem and the main robot against the game manual's weight limit; record it in version descriptions.
35. **Design review:**
36. Comments (right-click → Add comment, @tag teammates).
37. Branches for alternatives; Compare and Merge.
38. Version names per `onshape-release-and-versioning`. Mentor sign-off before manufacture.
39. **Manufacturing outputs:**
    - Drawings per part (`onshape-professional-drawings`).
    - DXFs from versions.
    - CAM Studio Manufacturing folders in each subassembly document, as 6329 does (plan availability UNVERIFIED).
40. **Onboarding and training:** CAD Basics → Onshape Fundamentals: CAD → CAD for Robotics pathway → FRCDesign.org lessons. Skip any custom-feature lessons. New members start in a sandbox copy of the T0 Template.

## Observed 6329 top-level structure (read-only)

- **Top assembly:** a single linked, fixed, version-pinned "Top-Level Assembly" holding only subsystem instances plus 4 Fastened mates.

- **Subsystems are separate documents at pinned versions,** using the instance menu's Open/Update linked document and Pin reference, so subteams work independently.
- **Part numbers:** `6329-YY-A####`, with the thousands digit = subsystem (A1000 Drivetrain, A3000 Intake, A5000 Shooter). This is the same idea as Postman's hundreds digit.
- **Names:**
- COTS subassemblies keep their vendor names (SDS MK5n, PDH, roboRIO 2.0, CANivore).
- Our own parts are named by function plus stock (`Front Tube 2"x1"x18.5"`).
- Leftover `Part 1 <n>` names were seen. Rename them.
- **Origin Cube:** 6329 places an "Origin Cube" datum at the top and in each subsystem. **We replace it natively:** a fixed layout or origin part plus named mate connectors from the T0 Template, so every subsystem mates to the same robot origin without a custom feature.
- **Access:** the linked subsystem documents could not be opened (permission error), so their Part Studio internals are unverified.

## Season timeline for CAD (INFERENCE, team practice)

| When | CAD milestone |
|---|---|
| Kickoff + 1–2 days | Concept doc: field elements, frame perimeter, crayon robot, extension limits |
| Week 1 | Layout V1.0 versioned; subsystem documents created from the T0 Template |
| Week 2–3 | Subsystem design reviews (CDR versions); weight budget check |
| Week 3–4 | Manufacturing versions; drawings and DXFs out |
| Before each event | Robot freeze version; BOM and spares list printed |

## Belt, chain and gear spacing without generators

- Compute center distances by hand (see `engineering-math-proofs`), e.g. an 80T 5 mm belt on 18T/36T pulleys gives C ≈ 5.186 in.
- Put the result in a Variable (`#arm_belt_cc = 5.186 in`) and dimension the layout sketch with it.
- Model the belt or chain as a native sketch plus extrude (or a simple sweep) only if it's needed for clearance.

## Do

- Keep one source of truth for robot geometry: the layout, derived by version.
- Number everything consistently from day one.
- Put the document directory and the tech binder in the main robot document.
- Measure from the origin, not from arbitrary tube faces.
- Keep subsystem assemblies rigid and the main assembly lightweight.

## Don't

- Don't put the whole robot in one document.
- Don't use the blog's custom features (Origin Cube, generators, Lighten).
- Don't copy-paste parts between documents; link versions.
- Don't let students start a subsystem without deriving the layout.
- Don't edit other teams' public documents; Copy them into your own workspace (if allowed) to study.

## Common mistakes

- **No numbering,** so the BOM and drawings can't be traced back to CAD.
- **The layout derived from a workspace,** so subsystems shift whenever someone edits it.

- **Top assemblies with hundreds of individually mated parts** instead of rigid subassemblies.
- **Missing materials,** so the weight estimate is fantasy.
- **Template features** that are custom features in disguise. Check the feature list of any template copied from the internet.
- **Folders shared without Link permission,** so teammates can't insert parts.

## Checks before done

- [ ] Season folder owned by the team; shared with Link permission.
- [ ] Concept, subsystem and main documents exist; links go by version.
- [ ] PS/A numbering and part numbers are applied.
- [ ] The T0 Template is versioned and contains no custom features.
- [ ] Every part has a material; the weight budget is under the limit with reserve.
- [ ] The main assembly has few instances and mates; subsystems are rigid.
- [ ] The document directory and tech binder are present.
- [ ] Training path shared; no custom-feature lessons assigned.

## Sources


- FRCDesignLib replaces MKCad: https://frcdesign.org/resources/frcdesignlib/ ; KrayonCAD: https://frcdesign.org/resources/krayoncad/
- FRCDesign document setup (Concept / Subsystem / Main Robot; Derive and Insert): https://frcdesign.org/best-practices/document-setup/
- Onshape blog, onboarding a robotics team (classroom ownership, folders, MKCAD, KrayonCAD, Standard content, comments, course path): https://www.onshape.com/en/blog/how-to-onboard-your-first-robotics-team
- FRC 6329 Dial release (top-level doc with assembly, reference sketches, directory, tech binder; shared reference sketches; measure to origin; ~4 M primitives): https://www.chiefdelphi.com/t/team-6329-the-bucks-wrath-2024-robot-reveal-dial/456061
- 6329 CAM Studio example: https://www.chiefdelphi.com/t/onshape-cam-studio-example/508160?page=2
- Learning paths: https://learn.onshape.com/learning-paths/introduction-to-cad , https://learn.onshape.com/learning-paths/onshape-fundamentals-cad
- Folders / linking / derived / configured properties: https://cad.onshape.com/help/Content/Document/folders.htm , https://cad.onshape.com/help/Content/Document/linking_documents.htm , https://cad.onshape.com/help/Content/PartStudio/derived.htm , https://cad.onshape.com/help/Content/PartStudio/configured_properties.htm
- Observed practice: the Onshape Explore Postman (PS100/A100 numbering, subsystem folders, reference sketch, `#DTST` variables, Master Sketches – Derived folder, 5-instance top assembly) and Jackhammer (instance folders by function, named mates, Render Studio and manual tabs).
- 6329 top-level assembly observed read-only in Onshape Explore; its linked subsystem documents returned a permission error, so their internals are unverified. The budget split and part-number scheme are INFERENCE.

# FRC robot design playbook

> **Summary (description):** Use when designing an FRC robot from a new game manual (kickoff), sizing a mechanism (motor, gearing, belts, beams, flywheel), organizing a large Onshape robot CAD, or reviewing a design for rules, weight, CG, current draw or reliability.

# FRC robot design playbook

> **MANDATORY FIRST STEP:** before choosing a concept or refining any design, run the `design-concept-gate` skill: list only the real constraints, challenge every self-imposed limit (height, part count, look, load case, print orientation), hand-calc 3-6 fundamentally different forms, and pick on numbers. No CAD or sizing until its decision table exists.

Reference notes and calculators live in `/workspace/frc-design-kb/` (backup copy in the bot's persistent store under `frc-design-kb/`). Read `README.md` there for the index.

## Before anything

- Check your memory for the user's standing rules (sign-ins, CAD tool limits, robot code, simulations, locked decisions, orders) and follow them over anything here.
- Free public sources only unless the user says otherwise. Don't predict new-season rules; read the manual.

## Evidence labels

Tag every number FACT (with a URL), INFERENCE or GUESS. Never invent a constant. If a spec has no public source, make it an input.

## Step 1: rules table (first ~3 h)

1. Download the manual PDF and run `pdftotext -layout`.
2. Grep and fill the table in `frc-rules-history.md` section 1: `^R10[0-9]` (weight, perimeter, height, extension), the bumper-weight rule, `BUMPER ZONE`, the motor table, the cost cap.
3. Compare with history: weight 120 lb (2015-18), 125 (2019-23), 125.5 (2024), 115 without bumpers (2025-26); 2026 perimeter at most 110 in, height at most 30 in, 12 in extension on one side only.
4. Pull field obstacle clearances into the layout sketch.

## Step 2: strategy (Karthik)

Strategic analysis first; look for a chokehold; list every way to score or deny scoring; cost/benefit each task; priority lists. Golden Rule #1: build within your limits. Golden Rule #2: 3 functions at 10/10 beat 5 at 6/10. Enumerate alliance permutations and pick a role. Source: https://robotics.choate.edu/wp-content/uploads/2019/12/KARTHIK-effective_first_strategies.pdf

## Step 3: cycle math

`python3 tools/cycle.py --dist-m D --vmax V --accel A --intake-s I --score-s S --period-s P --points X`. Rank tasks by points per second and ranking-point probability. Inputs are GUESSES until prototypes are measured.

## Step 4: requirements

Numeric requirements per function (pieces, seconds, heights, poses). Cap core functions at 3.

## Step 5: concepts and calculations (`tools/`, each has `--help`)

- Motor: `motor.py --motor krakenx60 --rpm R --current-limit A`. Size at the current limit, not stall.
- Drive: `gearing.py drive --motor-rpm 6000 --mk5i R3`. MK5i ratios come from YAGSL (secondary); verify on the SDS page.
- Belts and chain: `gearing.py belt|chain`. Even chain link counts. WCP center offsets: 0 to -0.003 in for gearbox stages, -0.015 to -0.020 in for rollers.
- Beams: `beam.py`. 6061-T6 E 68.9 GPa, yield 240 MPa, fatigue 97 MPa; design cycled parts for fatigue and use about 165 MPa near welds.
- Flywheel: `shooter.py spinup|drop|traj`. Drag-free trajectory is a lower bound; measure the shot-energy factor.
- Weight: `weight.py parts.csv --year YYYY` or `--limit-lb`. Keep a reserve.
- CG and tipping: `cg_tip.py` (a_tip = g*b/h). Check every pose.
- Run `cd tools && python3 -m unittest` first; all tests must pass before trusting outputs.

## Step 6: CAD (top-down, Onshape)

Master layout sketch (perimeter, bumpers, obstacles, game piece, every pose); Variable Studio for key dimensions; block CAD first (real power transmission and COTS geometry, simple plates), detail on top; COTS from FRCDesignLib; cross-check in-sketch belt/chain functions with `gearing.py`; check interference and extension rules per pose.

For a large robot (1,000+ instances), follow `onshape-large-robot-playbook.md`. Core rules: 1. Each moving body is one rigid subassembly; subsystem assemblies only define motion between bodies; the robot assembly inserts subsystem assemblies, never loose parts. 2. One document per subsystem: master layout, then subsystem documents, then the robot document. Derive only from the layout's version, never between subsystems. 3. Under 40 tabs per document (hard ceiling 100), under 10 workspaces; imports go in library documents. 4. About 10 unique parts or 250 features per Part Studio; split if any feature takes over 10 s; slow features last. 5. Every part has a material or a mass override. Check "parts missing mass = 0" at every review. 6. Version at every milestone; edit on branches, merge into a protected Main; update robot-assembly references on purpose. 7. No default names; use a part-number scheme (e.g. `TEAM-YY-P-SSNN`); no "Copy" or "FINAL" duplicates; old variants go to an archive folder or a version. 8. Insert each unique fastener once, then Replicate it; all bolt holes are Hole features. 9. Use simplified COTS models and a lightweight display state. 10. Mate solve under 1 s; prefer Fastened mates and Groups; minimize limits and tangent mates. 11. A named position for every robot state, with an interference check in each. 12. Joints that come apart often use through-bolts with nylock or rivet nuts; tap only walls of 1/8 in or more; riveted joints get at least one bolt.

Bot use of Onshape: read endpoints (elements, versions/workspaces, BOM, assembly definition with `includeMateFeatures=true`, mass properties, bounding boxes) from the signed-in browser. Any write (API or GUI) only with the user's OK, on a branch, after making a version. Only read documents the team owns.

## Step 7: budgets

- Weight under the limit with reserve, checked with bumpers.
- Current: WPILib example ~180 A sustained; 240 A for more than 1-2 s causes trouble. Brownout stages at 6.8 V and 6.3 V (roboRIO 2 default 6.75 V); re-check for the current controller.
- CG: tip acceleration above achievable drive acceleration in every pose.

## Step 8: reliability

FMEA from the function list in week 1 (RPN = S x O x D; fix every top-severity item). Designed fuses (shear pins, polycarbonate), impact guards, belt-skip prevention, WPILib wiring practices (pull-test, no hand-rotatable terminals, re-check battery and PDP bolts every 1-3 matches, shock-mount the radio). See `reliability.md`.

## Step 9: industrial design

Every part earns its place; structure doubles as function; electronics, breaker and battery easy to reach; material by function; 3D prints loaded along the layers. See `industrial-design.md`.

## Step 10: release V1 (end of the 72 h plan in `design-process.md`)

CAD version, parts list, drivebase DXFs, prototype plan with pass criteria, GUESS list for human review, decision log with sources.

## Simulation

For which simulation features fit each mechanism, see `simscale.md` and, once written, `frc-mechanisms-2017-2026.md` and `sim-app-feature-list.md`. Hand-check every simulation before trusting it.

## Common failures

Mechanisms before strategy; too many functions; gearing chosen on free speed; odd chain link counts; unconstrained extension poses; detail CAD before layout freeze; flat assemblies with imported copies and no masses; trusting simulations without a hand check; unsourced numbers.

## Champion reference points

2026 4414 (https://2026.team4414.com/) and 2025 2910 (https://frcteam2910.org/2025/05/07/2025-recap-post/): INFERENCE, both won on throughput, reliability and iteration rather than novelty.

# frc-subsystem-arms-wrists-and-pivots

**Summary (description):** Use when designing, sizing, modeling or reviewing an FRC arm, shoulder, wrist, or any powered pivot (intake pivot, shooter hood/pivot): dead vs live axle, superstructure, final-stage chain reduction, gearbox shock protection, counterbalance with gas springs, absolute encoders, and the Onshape layout. Mechanical only.

# FRC subsystem: arms, wrists and pivots

Labels: FACT (sourced, keys in /workspace/skill-drafts-frc/SOURCES.md) / INFERENCE / GUESS. Math: `engineering-math-proofs` (arm hold torque; Motors, gearing; Beams; Bearings, columns, shafts). Onshape GUI only, never FeatureScript.

## 1. Requirement questions

- [ ] Every target angle, plus the angle sweep time. Which positions are held for a long time?
- [ ] Arm length and mass at the tip (piece + end effector). (FACT [A1]: "Try to keep your arm under 20 lbs".)
- [ ] Does it cross over the robot (flip sides)? That moves backlash from one side to the other (FACT [A1]).
- [ ] Will it be hit, or drive into the field at speed? (Long-lever shock is the main killer, FACT [K].)
- [ ] Extension rule: in 2026 the reach must stay ≤ 12 in past the perimeter, one side at a time (R105/R106 [R1]).
- [ ] Is a wrist needed, or can geometry solve it? (Draw it in the master sketch first, FACT [A1].)

## 2. Architecture options

| Option | Pros | Cons | Example |
|---|---|---|---|
| Dead axle + chain/large sprocket | No torque through the axle; gearbox shielded by the final stage; strong | Needs an off-axis encoder drive | 7461 2023, 2910 2023 [A1][A2] |
| Live hex axle | Simple, encoder on the shaft | ½ in hex "can and will bend or shear" on long levers (FACT CD 421505) | Short, light wrists only |
| Belt final stage | Quiet, low backlash | 1323 stopped using belts to rotate mechanisms "because it ends poorly" (FACT [K]) | Light wrists |
| Gear final stage (sector) | Compact | 10DP teeth sheared under long-lever shock (FACT [K]) | Hoods (254 2020 printed sector [K]) |
| Low-mounted gearbox + chain up to joint | Lowers CG | Long chain runs need tension | 6328 2023 A-frame, 228:1 second joint, 64T MAXSpline sprocket [A3] |

## 3. Rules of thumb (FACT, CD arm guide [A1], stated by the author as "based mostly on vibes")

- Support towers 6–7 in apart. Over 8 in "gets sketchy" for bending of the dead axle.
- Dead axle: 7/8 in OD 6061 tube ($I \approx 0.23$ in$^4$ vs 0.095 for ½ in hex as quoted), bronze bushings in the arm, and concentric holes drilled through both walls in one setup.
- Final stage: #35 chain, small (12T) to large (52–54T+) sprocket. The bolts sit near the sprocket edge (¼-20), not near the axle.
- Ballpark: 160–200:1 total with 2 brushless CIM-class motors.

- Absolute encoder at 1:1 with the arm, backlash-free (REV Through Bore or CTRE CANcoder), driven off-axis by a pulley on the arm.
- Turnbuckles tightened by hand only; wrench-tightened ones broke on 7461.

## 4. Sizing math

1. Hold torque T = W·r_cg·cos θ from horizontal (eng-math arm hold torque). Use the worst θ.
2. Ratio from sweep time with a current limit (ReCalc Arm [A4]).
3. Gearbox shock check: gearbox output torque = joint torque ÷ final-stage ratio. Compare to the static rating: MAXPlanetary 4:1 stage 270 N·m, 9:1 stage 180 N·m (FACT, REV). Use a shock factor for crash loads (GUESS 3–5×; replace with measured current spikes if available).
4. Axle bending: a two-support beam with the arm load between them (eng-math Beams). Pass = no yield at the crash load, deflection small enough to keep the chain in mesh.
5. Gas-spring counterbalance: needed spring force F = W·r_cg/r_mount at the design angle. Buy 60–75 % of it (FACT [A1]) so gravity still keeps backlash on one side.

## 5. COTS parts

- Axles and bearings: 7/8 in 6061 round tube; McMaster bronze bushings; REV MAXSpline + MAXSpline sprockets; WCP/AndyMark #35 bearing-bore plate sprockets (FACT [A1]).
- Gearboxes: REV MAXPlanetary, WCP / VEXpro VersaPlanetary, TTB; custom 3-stage plates.
- Tensioners: inline turnbuckles or Spartan tensioners (FACT [A3]).
- Encoders: REV Through Bore (V2), CTRE CANcoder.
- Counterbalance: gas springs (McMaster, AndyMark), constant-force springs.

## 6. Onshape modeling approach

- Master sketch: pivot point, arm as a construction line with an angle dimension, tip circle sweep, frame, bumpers, the 12 in extension line, and the scoring targets. Draw a circle for the clearance at each contact point (FACT [A1]).
- Make the arm angle a variable `#armDeg` and build configurations `Stow / Intake / L2 / L4` (`onshape-variables-and-configurations`). Check interference in each.
- In the assembly use a revolute mate with limits at the hard stops; drive gas-spring geometry with a mate at each end + a slider (`onshape-assemblies-and-mates`).
- Sprocket center distance comes from eng-math Belts, chains (`mech-design-shafts-bearings-gears-belts`). Model the turnbuckle slot so tension can be set.
- Use Mass properties on the arm sub-assembly for W and r_cg, and recheck after every end-effector change.

## 7. Worked mini-example: 7461-style arm

- FACT [A1]: the 7461 beta arm needed an 83 lb gas spring at a 3 in mount radius with a 25 in CG; they bought 50 lb (60 %).
- INFERENCE: implied arm weight = 83·3/25 ≈ 10 lb → hold torque ≈ 10 lb × 25 in = 250 lb·in = 28 N·m.
- With a 12T → 54T final stage (4.5:1), gearbox output sees 28/4.5 ≈ 6.3 N·m static. With a 5× crash factor (GUESS) that is ≈ 31 N·m, well under a 180 N·m MAXPlanetary 9:1 stage. The chain stage is what keeps it safe.
- Without the chain stage (live hex at the joint), the same crash puts ≈ 140 N·m into the gearbox and bends the hex. That matches the reported failures (FACT CD 458372, 421505).

## 8. Failure modes and fixes

| Failure | Evidence | Fix |
|---|---|---|
| Planetary pins broke on fast stops | FACT CD 458372 | Chain + large sprocket; soft limits; hard stops on structure |
| ½ in hex bent or sheared | FACT CD 421505 | Dead axle, bigger tube |
| Teeth sheared on long levers | FACT CD 500435 | Chain final stage; wider face |
| Turnbuckle snapped | FACT [A1] | Hand-tighten; Spartan tensioner |
| Encoder slipped, so position drifts | INFERENCE | Keyed 1:1 drive; witness mark; check it on the pit list |
| Arm oscillates under the gas spring | FACT [A1] | Under-spec the spring to 60–75 % |

## 9. Gotchas

- Gas springs are nearly constant force, so torque varies with geometry. Plot torque vs angle (FACT CD synthesis [A5]).
- Assemble with the gas spring fully extended at the arm's max position so no one has to compress 50 lb by hand (FACT [A1]).
- Mass at the tip counts twice: hold torque AND inertia (INFERENCE).
- Wrists on an arm add their full mass at the tip. Budget them in `frc-weight-and-cg-management`.

## 10. Review checklist

- [ ] Hold torque and sweep time calculated at the worst angle, with a current limit.
- [ ] Gearbox output torque at a crash factor is below the rated static torque.
- [ ] Dead axle, bushings, concentric holes; towers ≤ 8 in apart or justified.
- [ ] Absolute encoder at 1:1, backlash-free, retained.
- [ ] Hard stops on structure at both ends; soft limits handed to programmers.
- [ ] Counterbalance (if any) 60–75 %; torque-vs-angle plot attached.
- [ ] Clearance checked in every configuration.

## Hand-off to programmers (numbers only; AI writes no robot code)

- [ ] Total ratio, absolute encoder ratio (1:1 preferred), and zero offset at a known pose.
- [ ] Hard-stop angles and the soft-limit margin; any no-go zones vs. other mechanisms.
- [ ] Gravity torque at horizontal (for feedforward) and the gas-spring torque curve.
- [ ] Arm MOI about the pivot from Onshape Mass properties.
- [ ] Max safe speed from the crash-load check in §7.
- [ ] Write these in the mechanism's spec doc, with units, and update them when hardware changes.

## Done when

Configurations show no interference, the shock check passes, the encoder reads 1:1 on the prototype with no slip after 50 cycles (GUESS gate), and the checklist is signed.

# frc-subsystem-climbers-and-endgame

**Summary (description):** Use when designing, sizing, modeling or reviewing an FRC climber or other endgame mechanism: hooks, telescopes, winches, ratchets and brakes, holding after power-off, hanging on one hook, swing and catch loads, and the Onshape layout against bar/rung geometry. Mechanical only.

# FRC subsystem: climbers and endgame

Labels: FACT (sourced, keys in /workspace/skill-drafts-frc/SOURCES.md) / INFERENCE / GUESS. Math: `engineering-math-proofs` (Motors, gearing, battery; Beams; Bearings, columns, shafts; Tipping and CG). Onshape GUI only, never FeatureScript.

## 1. Requirement questions

- [ ] Bar/rung geometry and heights (2026: rungs 1.66 in OD at 27 / 45 / 63 in, FACT [R1]). Which level is worth it in points vs cycle time?
- [ ] Climb time budget (seconds left in endgame after the last cycle).
- [ ] Robot weight on the hook: up to 135 lb with bumpers and battery (R408 [R1]).
- [ ] Must it hold after the match ends and power is cut? (Assume yes.)
- [ ] Can it share a motor or mechanism (elevator, shooter pivot)? FACT: Snoopy 2024 climbed with the shooter pivot; 254 2022 climbed with the elevator [K].
- [ ] Multiple attempts in a match (re-latch), or one shot?

## 2. Architecture options

| Option | Pros | Cons | Example |
|---|---|---|---|
| Telescope + static hooks | Compact; fast | Thin-wall tubes in compression/bending | 1619 2022 [K] |
| Winch + strap/rope | Huge reduction is easy; light | Needs a ratchet; spool wrap changes radius | 148 2018 3× 775pro, self + partner < 3 s [K] |
| Elevator does the climb | No extra motors | Elevator must carry ~robot weight; 254 2022 "500+ lbs" climb forces [K] | 254 2022 [K] |
| Pivot / arm climb | Shared mechanism | Big joint torques | Snoopy 2024 115.2:1 pivot [K] |
| Passive latch + gravity | Very reliable hold (INFERENCE) | Single attempt | Deep-cage era 2025, e.g. 2910 [K] |

## 3. Holding after power-off

- FACT [C1][C2]: use a ratchet/pawl on the winch shaft to stop back-drive; a small pneumatic cylinder or servo can lift the pawl for re-attempts.
- FACT [C2]: put the ratchet on the motor side of any dog clutch so releasing the clutch frees the winch without loading the pawl.
- FACT [C3]: pawl and ratchet teeth in steel; raked sawtooth; place the release path tangent to the loaded surface to minimize release force.
- INFERENCE (CD synthesis [C4]): motor brake mode is not a hold after power-off. Use a mechanical lock.
- COTS: WCP ratchet and pawl, SKU WCP-0405 (FACT [C1]).

## 4. Sizing math

1. Drum torque T = W·r_drum. Strap wrap grows r, so use the final (largest) radius for torque and the first for speed (INFERENCE).
2. Ratio from climb time with a current limit (eng-math Motors).
3. Hook and tube: full robot weight on ONE hook (FACT: 2022 traversal failures came from hanging on one hook [K]). Check bending and the shear-out at holes (eng-math Beams; Fasteners).
4. Static proof load 1.5–2× robot weight (GUESS; no public source found). Swing or catch loads are dynamic; use a measured spike or an impact factor ≥ 2 (GUESS).
5. Ratchet tooth and pawl pin: bearing stress and shear at proof load (eng-math Fasteners).
6. Telescoping tubes in compression: column buckling (eng-math Bearings, columns, shafts).

## 5. COTS parts

- Ratchets: WCP ratchet/pawl (WCP-0405) [C1]; one-way bearings (FACT: 254 2017 one-way-bearing PTO [K]).
- Gearboxes: REV MAXPlanetary, WCP / VEXpro VersaPlanetary, WCP climber kits. Check the static rating at the climb torque.
- Strap and rope: polyester or nylon strap; Dyneema (McMaster, WCP).
- Gas springs for deploy (FACT: 254 2022 30 lb gas spring [K]).
- Hooks: 1/4 in 6061 or 7075 plate, or steel for the contact zone (INFERENCE from the 2022 soft-alloy hook failure [K]).

## 6. Onshape modeling approach

- Master sketch: floor, rung circles at the true OD and heights from the manual, the robot envelope at the approach pose, the hook path, and the robot CG while hanging (`onshape-top-down-master-layout`).
- Configurations `Stowed / Reach / Latched / Hung` driven by the telescope or pivot variable (`onshape-variables-and-configurations`). Check the start height and extension limits in each.
- Model the hook throat around the rung OD with a clearance variable `#hookGap`. Check it with the rung at max tolerance.
- Static FEA on the hook and tube with full weight on one hook; read deflection and stress (`onshape-simulation-fea-sanity`).
- Mass properties with the robot hung: the CG must sit under the hook or the robot swings and peels off (INFERENCE).

## 7. Worked mini-example: winch climb to a 2026 rung

- W = 135 lb ≈ 600 N. 1.0 in drum (r = 0.0127 m) → T_drum = 7.6 N·m at the first wrap.
- Climb 18 in (GUESS) in ≤ 3 s → 0.15 m/s → drum 11.8 rad/s (113 rpm).
- 2× Kraken X60 at 25:1 (GUESS) → drum free ≈ 240 rpm. Load per motor = 7.6/(2 × 25 × 0.85) ≈ 0.18 N·m ≈ 9 A per motor (kT = 0.0194 N·m/A, FACT [K]). An easy, fast climb on paper.
- Proof: 2× → 15.2 N·m at the drum. The ratchet and every shaft adapter must carry that with the motors off (INFERENCE).
- FACT [K]: a VersaPlanetary shaft adapter failed after 12 good traverses and the housing spun, cutting about a dozen wires. Add a fatigue check and an anti-rotation tab on the motor.

## 8. Failure modes and fixes

| Failure | Evidence | Fix |
|---|---|---|
| Hook tubes sheared/bent hanging on one hook | FACT CD 404942 | Size for one-hook hang; thicker wall, not lightened 1/8 in (same-weight 1/16 in fared better in modeling) |
| Soft-alloy hook warped then broke | FACT CD 404942 | 7075 or steel contact zone |

| Failure | Evidence | Fix |
|---|---|---|
| Shaft adapter fatigue → housing spun | FACT CD 404942 | Anti-rotation; retire adapters by cycle count |
| Robot drops after disable | INFERENCE [C4] | Ratchet or brake |
| Pawl won't release under load | FACT [C3] | Tangent release path; release before load |
| Strap walks off the drum | INFERENCE | Flanged drum; guide |

## 9. Gotchas

- Test with the full-weight robot, battery and bumpers. A light practice frame hides failures.
- Climbs are rare in practice, so count cycles and inspect them at every event.
- Wires that cross the climber pivot or telescope need service loops.
- Read the endgame rules for touching partners, contact and protected zones before choosing a buddy climb.

## 10. Review checklist

- [ ] Rung geometry from the current manual in the layout.
- [ ] One-hook full-weight case checked; FEA deflection OK.
- [ ] Mechanical hold after power-off, demonstrated.
- [ ] Proof load test at the chosen factor (GUESS 1.5–2×) passed without permanent set.
- [ ] Climb time measured ≤ the endgame budget with margin.
- [ ] Cycle count logged; adapters and pawls on the inspection list.

## Hand-off to programmers (numbers only; AI writes no robot code)

- [ ] Drum radius at first and last wrap, ratio, and the climb distance per motor rotation.
- [ ] Ratchet engage/release actuation (solenoid or servo) and when it is safe to release.
- [ ] Hard-stop positions and current at the top; the stall current that means 'latched'.
- [ ] Climb time measured at full weight.
- [ ] Anything that must never move after latch (so the code leaves it in brake/disabled).
- [ ] Write these in the mechanism's spec doc, with units, and update them when hardware changes.

## Done when

The robot climbs at full weight within the time budget three times in a row (GUESS gate), holds 2 min after disable with no creep, passes proof load, and the CAD has the rung layout and configurations.

# frc-subsystem-drivetrain

**Summary (description):** Use when choosing, sizing, modeling or reviewing an FRC drivetrain: swerve vs tank/WCD, which COTS swerve module and ratio, frame tube and bellypan, traction and pushing current, and the Onshape layout of the drivebase. Mechanical only; no robot code.

# FRC subsystem: drivetrain

Labels: FACT = sourced (key in /workspace/skill-drafts-frc/SOURCES.md), INFERENCE = derived from facts, GUESS = starting value to replace with test data. Math lives in `engineering-math-proofs` (Motors, gearing, battery; Tipping and CG; Beams). Cite it, don't re-derive. Onshape is GUI-only: sketches, Part Studios, variables, configurations, assemblies. Never FeatureScript.

## 1. Requirement questions (answer before picking modules)

- [ ] Does the game reward lateral moves, fast scoring alignment or heavy defense? (FACT: swerve is "the most commonly used competitive drivetrain", FRCDesign [D1].)
- [ ] Field obstacles: bumps, ramps, a charge station, or a trench height limit?
- [ ] Target free speed and 0–max time. Will the robot push, or be pushed?
- [ ] Budget for 8 motors + 8 controllers + 4 modules (swerve), or 4–6 motors (tank)?
- [ ] Frame size against the 2026 limit: perimeter ≤ 110 in (R104 [R1]). Is there clearance for the bumpers and the extension side?
- [ ] Spare modules on hand, and the target module swap time (see `frc-competition-readiness`)?

## 2. Architecture options

| Option | Pros | Cons | When |
|---|---|---|---|
| COTS swerve (SDS, REV, WCP, TTB) | Holonomic; mature COTS; 8 motors give big pushing current | Cost, 8 motors, gear wear, more failure points | Default for competitive teams (FACT [D1]) |
| WCD / tank, 6 wheel drop-center | Simple, robust, cheap; resists lateral pushes | No strafing; slower alignment | Rookie or low-budget; heavy-defense games (INFERENCE [D6]) |
| KitBot tank (AM14U) | Arrives in the kit; known good | Heavy; slow alignment | First year, or a backup chassis |
| Mecanum | Strafes cheaply | Weak pushing; poor traction | Rarely competitive now (INFERENCE) |

## 3. COTS modules and ratios (FACT, vendor/YAGSL tables [D2][D3][D4])

| Module | Wheel | Drive ratios | Steer | Mass |
|---|---|---|---|---|
| SDS MK5i | 4 in | R1 7.03 / R2 6.03 / R3 5.27 | 26:1 | 4.1 lb bare / 6.1 lb ready |
| SDS MK4i | 4 in | L1 8.14 / L2 6.75 / L3 6.12 | 21.43:1 (150/7) | 6.0–6.3 lb |
| SDS MK4n | 4 in | L1 7.13 / L2 5.9 / L3 5.36 | 18.75:1 | — |
| REV MAXSwerve | 3 in | (45·22)/(pinion·15): 12T 5.50, 13T 5.08, 14T 4.71 | — | 3.8–3.97 lb |
| Thrifty Swerve | — | see vendor | 25:1 | 5.6–6.28 lb |
| - FACT [D2]: on MK5i only the pinion changes the ratio; the bevels are enclosed; 7075-T6 plates; 10-32 auxiliary holes; mounts top | | | | |

| Module | Wheel | Drive ratios | Steer | Mass |
|---|---|---|---|---|
| or bottom. | | | | |
| - FACT [D4]: the MAXSwerve upgrade kit (REV-21-3008) adds 15T/16T pinions. | | | | |
| - INFERENCE (eng-math v = ωf·r/G): free speed MK5i R2 + Kraken X60 = 6000/6.03 rpm × π·4 in ≈ 17.4 ft/s; MAXSwerve 14T + NEO ≈ 15.8 ft/s. Real top speed is lower (GUESS: 85–90 %). | | | | |
| - FACT [K]: 4414 2026 chose 7.67:1 "for lower current draw"; 254 2022 ran MK4i L3 at 18 ft/s. | | | | |

# 4. Sizing math (cite eng-math: Motors, gearing, battery)

1. Free speed from ratio and wheel (above). Choose by sprint distance, not peak speed: short fields favor lower ratios (INFERENCE).
2. Traction-limited force: F = μ·W. Tread μ: blue nitrile roughtop 1.19, pebbletop 1.24 (AndyMark, FACT [D5]); Colson ~1.0–1.2 (older team tests, FACT [D6b]).
3. Current at traction: T_motor = (F/4)·r_wheel/(G·η). Choose a stator limit just below slip (FACT, CD synthesis [D6]).
4. Tipping under acceleration: a_tip = g·d/h (eng-math Tipping and CG). Set the acceleration limit below this, and hand that number to the programmers.

# 5. Structure

- FACT [D1]: 2×1 in box tube, 0.196 in holes on a 0.5 in pattern; 1/8 in aluminum bellypan with module cutouts.
- FACT [D7]: with bumpers, "plenty of teams" run 2×1×1/16 frames. A 10-32 thread gets ~2 threads in 1/16 in wall, so use nuts or rivnuts, and don't crush the 2 in face. Some teams run 1/8 in on the front and back rails for impacts.
- FACT [K]: 254 2022 used an unpocketed 1/8 in bellypan to lower the CG; 1323 2025 used the tops of the swerve plates as structure.

# 6. Onshape modeling approach (GUI only)

- Start from the master layout (`onshape-top-down-master-layout`): frame rectangle, module centers, bumper outline (backing + padding stack), battery box, and the CG target.
- Variables: `#frameW`, `#frameL`, `#tubeWall`, `#moduleInset`. Make frame size a configuration (`onshape-variables-and-configurations`) so 26×26 vs 28×28 is one switch.
- Insert vendor module CAD as a linked document, pinned to a version (`onshape-release-and-versioning`). Don't remodel COTS parts.
- Mate modules with fastened mates to the frame mate connectors; group the frame (`onshape-assemblies-and-mates`).
- Check the perimeter with Measure on the frame sketch. Check mass and CG with Mass properties on the assembly (`frc-weight-and-cg-management`).

# 7. Worked mini-example: 2026 MK5i R2 on Kraken X60

- Robot 135 lb with bumpers → W ≈ 600 N. Blue nitrile μ = 1.19 → 179 N per wheel.
- Wheel torque 179 × 0.0508 m = 9.1 N·m → motor torque 9.1/(6.03 × 0.9 GUESS η) = 1.67 N·m.
- Kraken kT ≈ 7.09/366 = 0.0194 N·m/A (FACT motor table) → ≈ 86 A stator per drive motor at the slip point.
- Decision (INFERENCE): a stator limit around 80 A stays just under slip. Pass it to the programmers, and log a pushing test to confirm.

- Free speed 17.4 ft/s. Check with the CG: if h = 9 in and d = 11 in, a_tip = 9.81 × 11/9 = 12 m/s² > the traction limit μg = 11.7 m/s², so it won't tip from drive alone (INFERENCE).

## 8. Failure modes and fixes

| Failure | Evidence | Fix |
|---|---|---|
| Drive pinion worn out | FACT: MK4 14T pinions "done" after ~10 h without grease [CD 433367] | Grease schedule; spare pinions |
| Chewed gears on hard stops | FACT: 5 m/s → 0 in one loop [CD 472000] | Acceleration limit (programmers own it) |
| Bolt fell out; seized bevel bearing | FACT [CD 472000] | Threadlocker spec + torque stripe (`industrial-reliability-design-for-frc`) |
| Encoder wire unplugged | FACT [CD 472000] | Strain relief + latch, pull test |
| Brownout from a loose main crimp | FACT [CD 457396] | Tug test every crimp; inspection list |
| Frame rail dented by hits | FACT [D7] | 1/8 in front/back rails or corner brackets (`frc-defense-and-robustness`) |
| Carpet melt / tread wear | FACT [D6] | Current limits; tread swap interval |

## 9. Gotchas

- Choosing the top ratio for speed, then losing every push and browning out (INFERENCE).
- MK5i and MK4i hole patterns differ; check the frame before a mid-season module swap (GUESS — verify against vendor CAD).
- 1/16 in tube with tapped holes. Use nuts or rivnuts (FACT [D7]).
- Forgetting the bumper and battery weight in the ~135 lb with-bumpers check (R408 [R1]).

## 10. Review checklist

- [ ] Module and ratio chosen with the speed AND traction-current math written down.
- [ ] Frame perimeter is legal in Measure; bumper outline is in the layout.
- [ ] Every module is fastened to a plate or nut and accessible for a ≤10 min swap (GUESS gate, `frc-competition-readiness`).
- [ ] Grease points and pinion spares are on the pit list.
- [ ] Stator/supply limits and the acceleration limit are handed to the programmers as numbers.
- [ ] CG height and a_tip are checked with Mass properties.

## Hand-off to programmers (numbers only; AI writes no robot code)

- [ ] Gear ratios (drive, steer), wheel diameter measured worn and new, and free speed.
- [ ] Stator current limit from the traction-current calc, and the supply limit for brownouts.
- [ ] Max accel before wheel slip and before tipping (a_tip), including at the extended CG.
- [ ] Module offsets and encoder zeroing procedure; bumper-to-bumper size for path planning.
- [ ] Robot mass with battery and bumpers (weighed, not CAD).
- [ ] Write these in the mechanism's spec doc, with units, and update them when hardware changes.

## Done when

The layout, configuration and assembly are built; the math is in the design notes; the checklist passes; and a pushing test log confirms the slip current (or is scheduled).

# frc-subsystem-elevators-and-telescopes

**Summary (description):** Use when designing, sizing, modeling or reviewing an FRC elevator or telescoping arm: cascade vs continuous rigging, stage count and overlap, bearing blocks, chain/rope/belt choice, gear ratio and time-to-height, springs, stiffness against skip, and the Onshape layout. Mechanical only.

# FRC subsystem: elevators and telescopes

Labels: FACT (sourced, keys in /workspace/skill-drafts-frc/SOURCES.md) / INFERENCE / GUESS. Math: `engineering-math-proofs` (Motors, gearing, battery; Beams; Belts, chains; Cable lengths). Onshape GUI only, never FeatureScript.

## 1. Requirement questions

- [ ] Highest and lowest scoring heights, measured from the floor to the end effector.
- [ ] Starting height limit (2026: ≤ 30 in total, R107 [R1]) → collapsed height → stage count.
- [ ] Payload mass (end effector + piece + any arm on the carriage).
- [ ] Time-to-height target (from cycle math in `frc-robot-design-playbook`).
- [ ] Does the elevator also climb? (Cascade puts about 2× force on the first stage, FACT [E3].)
- [ ] Vertical or tilted? Is there a wrist or arm on the carriage?

## 2. Architecture options (FACT [E1][E2][E3])

| | Cascade | Continuous |
|---|---|---|
| Motion | Every stage moves the same amount relative to the previous one | The lowest-friction (last) stage moves first; needs a hard stop at each stage |
| Force | First stage carries ~2× (FACT [E3]); higher tensions (FIRST [E4]) | Less tension (FIRST [E4]); jams more easily |
| CG | Higher mid-travel | Lower until the upper stages move |
| COTS | WCP GreyT, Thrifty Elevator, FRCDesign sample | Mostly custom (1678/3847 2023, 4414 2023, 111) |
| Adding stages | Hard past 3 | Easy: run the line through another stage |
| Typical rigging | Chain on stage 1, Dyneema after | Dyneema on pulleys, or timing belt (often in-tube) |
| - Telescope = an elevator on its side or at an angle. Same rules, plus bending of the stages under the payload moment (eng-math Beams). | | |

## 3. Structure rules

- FACT [E3]: most use 2×1 or 1×1 tube on bearing blocks. Keep ≥ 2 bearing blocks in contact per side between stages. Never leave a stage without a top crossbar.
- FACT [E3] rule of thumb: stage overlap ≈ 20 % of stage height (30 in stage → 6 in overlap). Treat it as a starting GUESS and check sway on the prototype.
- FACT [K]: 4414 2023 moved to 0.75 in bearings after small bearings cracked or wore into the tubes.
- FACT [K]: the most-cited failure is chain/belt skip from a bending crossbar or bowing end points (CD 347386, 1323 in CD 495375). **Size for stiffness, not just strength.**

- FACT [K]: #25 chain skips or derails "so much easier than 35"; the GreyT skip was fixed with #35.

## 4. Sizing math

1. Load: F = m·(g + a) at the motor-side member; multiply by the cascade factor for each stage (eng-math free-body).
2. Pulley/sprocket pitch radius from tooth count: HTD5 r = N·5 mm/(2π); chain r from eng-math Belts, chains.
3. Time-to-height: integrate the motor curve with a current limit (eng-math Motors; ReCalc Linear Mechanism [E5]).
4. Hold current at height: I = m·g·r/(G·η·n·kT). Keep it low enough to hold at the top for many seconds (INFERENCE).
5. Stiffness: crossbar deflection under the belt/chain tension (eng-math Beams). The pass rule is no center-distance change big enough to skip (see `mech-design-shafts-bearings-gears-belts`).
6. Springs: constant-force springs balance carriage weight. Size a little under full balance so the system still falls predictably (FACT CD 131229 summary [E6]; INFERENCE).

## 5. COTS parts

- Kits: WCP GreyT Elevator (1–3 stage cascade, 3/4 in bearings, 1/2 in hole spacing [E7]); The Thrifty Elevator (1–2 stage); WCP/TTB bearing blocks (inline or face-clamping).
- Rigging: #25H / #35 chain + Thrifty #25 comb hardpoints; Dyneema with WCP ratchet blocks; HTD5 Kevlar or fiberglass belts (Gates, WCP, TTB).
- Motors: 1–2 Kraken X60 (WCP) or NEO/Vortex (REV) on WCP or MAXPlanetary gearboxes (FACT [E3]).
- Springs: constant-force springs (McMaster-Carr, Vulcan), used by 4414 2023 and 3847 2023 [K][E3].

## 6. Onshape modeling approach

- Master sketch: floor line, collapsed and extended heights, each stage as a rectangle, overlap dimension, hard-stop lines, and the carriage-to-arm envelope (`onshape-top-down-master-layout`).
- Variables: `#stageH`, `#overlap = 0.2*#stageH`, `#travel`, `#nStages`. Make `Collapsed / Mid / Extended` a configuration, driven by one variable per stage (`onshape-variables-and-configurations`).
- In the assembly use slider mates per stage, limits set to the hard stops, and a linear mate relation (or a configuration) for cascade ratios (`onshape-assemblies-and-mates`). Lesson from the 2026-10 desk build: relations between hidden/grouped sliders can error ("One or more mates in this relation are invalid"); keep slider mates on visible, ungrouped instances.
- Measure chain and rope lengths with eng-math Cable lengths at both extremes. Lengths that differ mean a tensioner is needed.
- Run a static FEA sanity check on the top crossbar with the rigging load and read deflection, not just stress (`onshape-simulation-fea-sanity`).

## 7. Worked mini-example: 2-stage continuous belt elevator, 1 m travel

- Payload 8 kg (GUESS), 22T HTD5 pulley → r = 22·5/(2π) = 17.5 mm. 2× Kraken X60, 60 A stator limit (GUESS), η = 0.85 (GUESS).
- Point-mass sim with a motor curve (INFERENCE):

| G | Time to 1 m | Hold current / motor |
|---|---|---|
| 4:1 | 0.40 s | 10.4 A |
| 5:1 | 0.48 s | 8.3 A |
| 6:1 | 0.57 s | 7.0 A |

| G | Time to 1 m | Hold current / motor |
|---|---|---|
| 8:1 | 0.74 s | 5.2 A |

- Choose ~5–6:1. FACT [K]: 4414 2023 ran 5.6:1 on a belt continuous elevator. The sim ignores battery sag and acceleration limits, so expect real times 1.5–2× longer (GUESS); verify against a logged run.
- The top crossbar carries roughly 2× belt tension. Check that its deflection stays below the skip threshold before cutting.

## 8. Failure modes and fixes

| Failure | Evidence | Fix |
|---|---|---|
| Chain/belt skip | FACT CD 347386, 495375 | Stiffen crossbars and end points; #35 chain; more wrap |
| Rivets in bearing mount sheared; sprocket teeth chiseled | FACT CD 484418 | Bolts instead of rivets at load paths; steel sprockets |
| Rigging pulled out by an alliance partner | FACT 1323 2018 → internal rigging 2019 [K] | Route rope inside tubes; cover the exposed runs |
| Motor stalled holding a heavy carriage | FACT 1323 2025 [K] | Springs, or a lower ratio check against hold current |
| Bearings wore into the tube | FACT 4414 2023 [K] | Bigger bearings; inline blocks |
| Fatigue of a fuse screw | FACT: 4-40 fuse screws replaced every event [K] | Plan the fuse and stock spares |

## 9. Gotchas

- A continuous elevator needs a hard stop on every stage or the order of motion changes (FACT [E3]).
- Cascade tension multiplies down the stages. Size the first-stage rigging and crossbar for it.
- Rope stretch changes the zero. Add a re-tension step to the pit checklist (INFERENCE).
- Tilting elevators add a bending load on the bearing blocks that a vertical prototype never saw.

## 10. Review checklist

- [ ] Collapsed height is legal; extended height meets the top goal with margin.
- [ ] Overlap ≥ the chosen value at full extension; ≥ 2 blocks in contact per side.
- [ ] Crossbar deflection is checked; chain size and wrap are justified.
- [ ] Ratio chosen from the time-to-height and hold-current table.
- [ ] Every stage has a hard stop, a soft-limit height for programmers, and a sensor zero.
- [ ] Rigging is protected from partners and field elements.
- [ ] Re-tension procedure is written down.

## Hand-off to programmers (numbers only; AI writes no robot code)

- [ ] Ratio, drum or pulley pitch radius, and meters per motor rotation.
- [ ] Hard-stop heights (bottom, top), with the soft-limit margin from the layout.
- [ ] Gravity feedforward (hold current from §7) and the spring force curve if springs are used.
- [ ] Max height at which full-speed driving is safe (from the tipping check).
- [ ] Homing method (limit switch or hard-stop current) and where the sensor is.
- [ ] Write these in the mechanism's spec doc, with units, and update them when hardware changes.

## Done when

The layout and configurations are built, the ratio table is in the notes, the FEA shows deflection under the skip threshold, and a logged full-speed run matches time-to-height within 30 % (GUESS) with no skip.

# frc-subsystem-intakes

**Summary (description):** Use when designing, sizing, modeling or reviewing an FRC ground or station intake (over-the-bumper, under-the-bumper, fixed, pivoting or linkage): roller speed, compression, roller material, deploy actuator, impact survival, and the Onshape layout. Mechanical only; no robot code.

# FRC subsystem: intakes

Labels: FACT (sourced, keys in /workspace/skill-drafts-frc/SOURCES.md) / INFERENCE / GUESS. Math is in `engineering-math-proofs` (Motors, gearing; Belts, chains). Onshape GUI only, never FeatureScript.

## 1. Requirement questions

- [ ] Game piece: shape, size, mass, squishiness, and how much it varies (manual + a measured batch).
- [ ] Where pieces are taken: floor, station, or both? From any orientation?
- [ ] "Touch it, own it" or centering needed? How wide must the capture zone be?
- [ ] Extension budget: 2026 allows 12 in, one side at a time (R105/R106 [R1]). Does the intake share that side with the shooter?
- [ ] Hand-off target (indexer, end effector) and the hand-off speed.
- [ ] Will the intake be hit by defenders and walls? (Assume yes. FACT: it "lives outside the frame" [I1].)

## 2. Golden rules (FACT, Torrance/254 via FRCDesign [I1])

1. Roller surface speed ≥ 2× robot max speed.
2. Maximize grip.
3. Squishy piece → rigid roller; hard piece → compliant roller.
4. Maximize intake width.
5. Build it robust.
6. Sense the game piece.

## 3. Architecture options

| Option | Pros | Cons | Example |
|---|---|---|---|
| Over-the-bumper (OTB) pivot | Wide; deploys past the bumper; simple | Exposed to hits; uses the extension budget | 1678 2022 linkage [I2] |
| Under-the-bumper (UTB) | Protected; no extension | Needs ground clearance and a bumper gap; low pieces only | Snoopy 2024, 4522 2024 [K] |
| Four-bar / linkage deploy | Controlled path; hard stops take impact | More parts | 254 2020 polycarbonate 4-bar [K] |
| Fixed roller | Fewest parts | Narrow reach | Station-fed games |
| Flexible polycarb fingers | Rides over pieces | Fatigue cracks at holes | 1619 2022 [K] |

## 4. Sizing math

- Surface speed: $v = \omega \cdot r$. Target $v \geq 2 \cdot v_{robot}$ (rule 1). FACT: 1678 2022 used about 2.5× [I2]; 254 2017 ran about 40 ft/s [K].
- Ratio: $G = \omega_{free} \cdot r / v$ (eng-math Motors, gearing). Check stall current with the piece jammed (FACT: an indexer stalled at 40 A, CD 516076).

- Compression: start at ¼–½ in (FACT, LearnFRC [I4]) and tune on the prototype. FRCDesign prefers grip tape to adding compression (FACT [I3]).
- Deploy pivot: size the hold torque with eng-math arm hold torque. Use a chain final stage so the gearbox is shielded from shock (FACT, 1678's 32T plate sprocket on a 30:1 Falcon [I2]).

## 5. COTS parts (vendor named; check current catalogs)

- Rollers: WCP/REV/TTB 1.125–2 in polycarbonate tube on dead axles (FACT, FRCDesign rollers lesson [I3]); grip tape; compliant wheels (AndyMark, WCP); AndyMark Stealth wheels.
- Drive: Kraken X60 (WCP), NEO / Vortex (REV), Minion (CTRE); MAXPlanetary (REV); WCP or TTB belts/pulleys.
- Structure: ¼ in polycarbonate or 1/8–3/16 in aluminum plates; 7075 plates where hits are worst (FACT, 4414 2023 [K]).
- Sensing: beam break, CANrange, or a Banner sensor (FACT, 1323 2025 used Banner + CANdi [K]).

## 6. Onshape modeling approach

- In the master sketch: bumper outline, the 12 in extension line, frame, the piece circle at its worst-case size, and the deploy pivot point (`onshape-top-down-master-layout`).
- Lay out the linkage or pivot in one sketch with driving dimensions only. Sweep it with a configuration `Stowed / Deployed / Hand-off` (`onshape-variables-and-configurations`) and check interference at each state.
- Model plates in context of the layout (`onshape-in-context-and-master-model`); rollers are simple extrudes with a `#rollerOD` variable.
- Belt/chain center distances come from eng-math (Belts, chains), not by eye (`mech-design-shafts-bearings-gears-belts`).
- Mass properties on the intake assembly; keep the far end light (`frc-weight-and-cg-management`).

## 7. Worked mini-example: roller ratio for a MAXSwerve 14T robot

- Robot free speed 15.8 ft/s (see `frc-subsystem-drivetrain`) → roller ≥ 31.6 ft/s = 9.63 m/s.
- 2 in roller (r = 0.0254 m) → ω = 379 rad/s = 3620 rpm.
- Kraken X60 free 6000 rpm → G = 6000/3620 ≈ 1.66:1.
- FACT [I3]: FRCDesign's rollers lesson runs a Kraken at 1.6:1. INFERENCE: consistent, and 1.6:1 gives ≈ 2.1× robot speed at free speed. Under load it will be slower, so check with a prototype rather than a stopwatch on free spin.
- Pivot: 1678-style 30:1 + chain to a 32T plate sprocket. Hold torque per eng-math; size the hard stop so it, not the chain, takes impact (FACT, 254 2020 hard stops [K]).

## 8. Failure modes and fixes

| Failure | Evidence | Fix |
|---|---|---|
| Dead axle bent ~15° by a robot corner | FACT CD 468927 | Larger tube axle; guard the roller ends |
| 1/8 in polycarb shattered; aluminum bent | FACT CD 468927 | Aluminum-reinforced polycarb; 7075 plates |
| Bearings popped out on side hits | FACT CD 515856; 1678 screws+washers [I2] | Retain with washers or flanged bearings |
| Belt sides out of time after hits | FACT CD 515856 | Drive both ends with a torque-transfer shaft [I2][I4] |
| Screw ripped through 1×1 tube | FACT CD 515856 | Through-bolt with a backing plate or wider tube |
| Lock collars slipped | FACT CD 515856 | Shaft shoulders/retaining rings; threadlocker on collar screws |
| Polycarb fingers crack at holes | INFERENCE from 1619/CD 468927 | Radius every inside corner; keep holes away from bend zones |

## 9. Gotchas

- Rollers spinning fast in free air but stalling on the piece: test under load (INFERENCE).
- A wide roller driven from one end twists and skips. Drive both ends (FACT [I4]).
- The deploy side and the shooter side together can break the one-side extension rule (R106 [R1]).
- Grip tape wears. Put it on the pit inspection list.

## 10. Review checklist

- [ ] Surface speed ≥ 2× robot speed under load, measured.
- [ ] Capture width and compression set from a prototype with worn and new pieces.
- [ ] Every bearing is retained; every collar has backup retention.
- [ ] Hard stops take the deploy impact; the gearbox sits behind a chain stage.
- [ ] Wall-ram test and defense-bot hits done (FACT: "Ram it into a wall" CD 515856).
- [ ] Sensor sees the piece in all orientations.
- [ ] Swap time ≤5 min with 2–4 bolts (GUESS gate; FACT 1002 two-bolt swap [K]).

## Hand-off to programmers (numbers only; AI writes no robot code)

- [ ] Roller ratio and the surface speed target (≥ 2× drive speed, per §2).
- [ ] Deploy pivot ratio, hard-stop angles, and the absolute encoder offset.
- [ ] Stall-current limit that clears a jam without cooking the motor.
- [ ] Sensor location and what 'piece acquired' means physically.
- [ ] Reverse/eject direction and speed for jams.
- [ ] Write these in the mechanism's spec doc, with units, and update them when hardware changes.

## Done when

The prototype hits the target speed under load, the CAD is in the master layout with stow/deploy configurations, the checklist passes, and the intake survives a wall ram with no permanent set.

# frc-subsystem-shooters-and-flywheels

**Summary (description):** Use when designing, sizing, modeling or reviewing an FRC flywheel shooter, hood, or turret: exit velocity, wheel size and RPM, flywheel inertia and recovery between shots, spin, compression, wheel expansion, belts, feed/indexer hand-off, and the Onshape layout. Mechanical only; shot maps and code belong to programmers.

# FRC subsystem: shooters and flywheels

Labels: FACT (sourced, keys in /workspace/skill-drafts-frc/SOURCES.md) / INFERENCE / GUESS. Math: `engineering-math-proofs` (Motors, gearing, battery; Belts, chains; Bearings, columns, shafts). Onshape GUI only, never FeatureScript.

## 1. Requirement questions

- [ ] Shot distances and target height/size → required exit speed and angle range. (FACT: FRCDesign's reference design needed > 12 m/s for its game [S1].)
- [ ] Fire rate: single shots or bursts? (Burst rate sets the flywheel inertia.)
- [ ] Piece: mass (2026 FUEL 0.448–0.500 lb [K]), compression stiffness, variation (worn vs new).
- [ ] Spin wanted (backspin for range/stability) or none (to reduce bounce-outs)? FACT: 254 chose differently by game (2017 backspin, 2020 2 in top wheel for backspin, 2022 matched speeds) [K].
- [ ] Fixed, adjustable hood, pivot, or turret? Shoot while moving?
- [ ] Space and the extension rule (R105–R107 [R1]).

## 2. Architecture options

| Option | Pros | Cons | Example |
|---|---|---|---|
| Single wheel + hood | Simple; one motor group | Exit ≈ half surface speed (hooded, no-slip model [S2]) | 254 2022 (9 in wheel) [K] |
| Dual wheel (top/bottom) | Exit ≈ surface speed; spin set by speed ratio | Two groups to control | 254 2020 4 in + 2 in; Snoopy 2024 7:10 differential [K] |
| Fixed hood | Fewest parts | Narrow range | 254 2017 14° hood [K] |
| Adjustable hood | Range from one location | Hood backlash spreads shots | 254 2020 printed sector gear [K] |
| Turret | Aim without turning the drive | Mass, wiring, a large bearing | 254 2022 12 in bearing ±120° [K]; 4414 2026 55.7:1 [K] |

## 3. Sizing math

1. Exit speed. Hooded single wheel, no slip at exit: $\omega_{wheel} = 2 \cdot v_p / r_w$ (FACT, Brettle analysis [S2]). Dual wheel: $v_p \approx$ surface speed (INFERENCE). Teams often run faster, into slip, and get more range (FACT [S2]).
2. Ratio. Put the operating speed near the motor's peak-power point, about ½ free speed under load, for fastest recovery (FACT: 254 2022 sized at "motor peak-power-point" [K]; Brettle [S2]).
3. Energy per shot $\geq \frac{1}{2} \cdot m \cdot v_p^2$. Brettle shows mechanical efficiency < 50 %, so the flywheel gives up ≥ 2× the piece's KE (FACT [S2]).
4. Speed drop: $\omega_2 = \omega_1 \cdot \sqrt{(1 - E_{shot}/E_{fly})}$, with $E_{fly} = \frac{1}{2} \cdot I \cdot \omega_1^2$ (INFERENCE from energy balance). Brettle suggests sizing I for a ratio $\omega_1/\omega_2 \approx 1.5$ as a first pass (FACT [S2]).
5. Spin-up and recovery time: ReCalc Flywheel (FACT [S3]). More inertia means a smaller drop but a longer spin-up (FACT [S1]).

6. Hoop stress / growth of the wheel at RPM. Coverings expand (FACT [K]).

## 4. COTS parts

- Wheels: 4 in is common (FACT [S1]); AndyMark Stealth, Colson, solid rollers. Avoid compliant or treaded wheels at high speed because they expand (FACT [S1]).
- Flywheels: brass or steel inertia wheels (WCP, TTB, AndyMark); FRCDesign used two 4 in brass flywheels at 4:3 [S1]; 4414 2026 used 0.25 in copper [K].
- Motors: 2 CIM-class brushless, e.g. Kraken X60 (WCP), Vortex (REV) (FACT [S1]); 4414 2026 used 4× X44 [K].
- Power transmission: belts. Pulleys > 24T on the flywheel and high tooth engagement (FACT [S1]).
- Turret bearings: large thin-section or X-contact bearing (FACT: Snoopy 2024 X-contact + lower bearing [K]).

## 4b. Feed and hand-off (the indexer sets consistency)

- [ ] Feed the piece at a repeatable speed and position. Inconsistent feed shows up as range spread (INFERENCE).
- [ ] Hold the next piece until the flywheel is back at speed (programmers own the gate; mechanical provides the sensor).
- [ ] Put a sensor at the hand-off (beam break or ToF) so "piece staged" is a real signal.
- [ ] Keep compression the same across worn and new pieces: rigid backstop, measured gap, not a flexible plate (INFERENCE).
- [ ] FACT [K]: 254 2020's serializer ran constantly with a pneumatic "Chock" to stage balls; on a jam the driver alternated direction. Design a reverse path.
- [ ] Indexer stall current: a stalled indexer at full hopper pulled 40 A (FACT CD 516076). Budget for it.

## 5. Onshape modeling approach

- Master sketch: the piece path from the indexer to the exit, piece circle at contact (with compression as a dimension `#compress`), wheel circles, the hood arc, and the exit angle line (`onshape-top-down-master-layout`).
- Make `#compress`, `#wheelD` and `#hoodDeg` variables; build configurations for the hood angles or wheel sizes you prototype (`onshape-variables-and-configurations`).
- Model the flywheel at true material (brass, steel). Read I about the axis from Mass properties on the rotating group, and use that I in the energy math rather than a guess.
- Belt center distances from eng-math Belts, chains. Pocket plates only after the FEA sanity check on the bearing plates with belt tension.

## 6. Worked mini-example: hooded 4 in shooter, 12 m/s

- $r_w$ = 2 in = 0.0508 m → $\omega = 2 \times 12/0.0508 = 472$ rad/s ≈ 4510 rpm (INFERENCE, model [S2]).
- Kraken X60 free 6000 rpm at a 4:3 reduction → 4500 rpm wheel free speed. That matches FRCDesign's 4:3 pick (FACT [S1]), but at free speed only; under load it will run below, so ReCalc it.
- Piece mass: 2026 FUEL 0.448–0.500 lb, so use 0.227 kg (FACT [K]). $KE = ½ \times 0.227 \times 12^2 = 16.3$ J; the flywheel gives up ≥ 2×, about 33 J (FACT bound [S2]).
- With $I = 0.002$ kg·m² (GUESS; read the real one from Onshape Mass properties), $E_{fly} = ½ \times 0.002 \times 472^2 = 223$ J → $\omega_2 = 472 \cdot \sqrt{(1 - 33/223)} = 436$ rad/s, an 8 % drop ($\omega_1/\omega_2 \approx 1.08$).
- Decision (INFERENCE): 1.08 is gentler than Brettle's 1.5 first pass, so less inertia would spin up faster. Check the burst rate in ReCalc before removing mass.

## 7. Failure modes and fixes

| Failure | Evidence | Fix |
|---|---|---|
| Roller covering expanded "dangerously" at RPM | FACT 1690 2024 [K] | Rubber vulcanized on aluminum hubs; solid wheels |
| Wheel expansion | FACT 254 2016 safety wire [K] | Safety wire or a rigid hub |
| Long roller bent → belt skip | FACT 1690 2024 steel "banana" plate [K] | Stiffen; shorter span; mid support |
| RPM drop between shots | INFERENCE; 4414 added copper mass [K] | More I, or more motors; gate the feeder on RPM |
| Inconsistent range from worn pieces | INFERENCE | Test the shot map with worn and new pieces |
| Hood backlash | INFERENCE | Spring-preload the hood to one side |

## 8. Gotchas

- Free-speed ratio math always looks right. The loaded speed is what shoots (INFERENCE).
- Spin changes range a lot. Decide it with a prototype, not theory (FACT 254 2017: an earlier test "incorrectly suggested ball spin was bad" [K]).
- A turret needs a cable wrap or service loop. Count it in the ±range.
- Guard spinning wheels from fingers and wires (pit safety).

## 9. Review checklist

- [ ] Exit speed requirement written with source distances.
- [ ] Operating speed near peak power under load; ReCalc spin-up and recovery attached.
- [ ] Real I read from Onshape Mass properties; drop per shot computed.
- [ ] Wheels and coverings rated (or tested) at 1.25× operating RPM (GUESS gate).
- [ ] Belts on pulleys > 24T with good wrap; rollers stiff enough not to bow.
- [ ] Hood and turret backlash handled; wire wrap designed.
- [ ] ~100-shot test across distances done before the shot map is trusted (FACT: 1690 used ~100 datapoints [K]).

## Hand-off to programmers (numbers only; AI writes no robot code)

- [ ] Flywheel ratio, wheel diameter, and the measured I from Onshape Mass properties.
- [ ] Target RPM range and the ready-to-fire tolerance that keeps range spread acceptable.
- [ ] Hood ratio, travel limits, and the encoder offset; turret limits and cable-wrap range.
- [ ] Measured RPM drop per shot and recovery time from the prototype.
- [ ] The shot map datapoints (distance, RPM, hood) as the prototype measured them.
- [ ] Write these in the mechanism's spec doc, with units, and update them when hardware changes.

## Done when

The prototype hits the required exit speed under load, burst recovery meets the fire-rate target, the coverings pass the over-speed test, and the CAD has the layout plus configurations.

# frc-systems-engineering

**Summary (description):** Use at kickoff and at every phase gate of an FRC season to turn game strategy into numbered, testable requirements, stage-gate reviews (SRR/PDR/CDR-style), and a technology-readiness call on every mechanism before CAD or fabrication is committed.

# FRC systems engineering (strategy → requirements → gates)

> **MANDATORY FIRST STEP:** before choosing a concept or refining any design, run the `design-concept-gate` skill: list only the real constraints, challenge every self-imposed limit (height, part count, look, load case, print orientation), hand-calc 3-6 fundamentally different forms, and pick on numbers. No CAD or sizing until its decision table exists.

Labels: FACT (source), INFERENCE, GUESS. Source keys S# (teams) and I# (industry) are listed in `/workspace/frc-design-kb/top-teams.md` (store backup under `frc-design-kb/`). Tools are in `/workspace/frc-design-kb/tools/`. This is the umbrella skill; frc-mechanism-design-review, frc-dfm-manufacturing, frc-reliability-testing, frc-prototyping-and-iteration, frc-weight-and-cg-management and frc-competition-readiness plug into its gates.

## Why

- FACT [S11]: 254 traces goal → ranking-point table → robot qualities → subsystem requirements with **time gates** (intake <0.5 s, elevator <1 s, climb engage <1 s).
- FACT [S14]: 2056: overestimating abilities and resources is the single most harmful thing; 3 functions at 10/10 beat 5 at 6/10.
- FACT [I1][I2]: NASA holds SRR → PDR → CDR; CDR asks whether the design is mature enough to start fabrication.
- FACT [I13]: the V-model pairs every requirement with a verification. Verification = built it right; validation = built the right thing.

## Step 1: Requirements (Day 0–1) → SRR

Run a rules test first [S18]. Then fill one row per function:

| ID | Need/Want/Nice | Requirement (numeric) | Rationale | Verify by (test/analysis/inspect) | Owner |
|---|---|---|---|---|---|
| R-INT-1 | Need | Acquire game piece from floor in ≤ __ s from contact | cycle model | timed test, 20 trials | |
| R-ELV-1 | Need | Full travel ≤ __ s, overshoot ≤ __ in | cycle model | 120+ fps video | |
| R-SYS-1 | Need | Robot mass ≤ limit − reserve | rules | scale | |

- [ ] Every requirement has a number, unit and verification method. "Fast" or "reliable" alone fails.
- [ ] Expected-value gate per scoring option: EV = P(success) × points ÷ cycle time. FACT [S16]: 971 went for the top hoop only at ≥67% accuracy, its break-even.
- [ ] Sort into Needs/Wants/Nice [S18]; only Needs get resources before PDR.
- [ ] Capacity check. FACT [S14]: teams assume ~2× real capacity. GUESS: plan on 50% of estimated build hours.

- [ ] Separate functional from non-functional requirements (mass, serviceability, cost, inspection) [S13] [S14].

**SRR exit gate:** 100% of Needs carry numbers; `tools/cycle.py` shows the set reaches the strategy target; rules-compliance list checked by two people.

## Step 2: Concepts and TRL (Day 1–5) → PDR

| FRC-TRL | Meaning | Evidence |
|---|---|---|
| 3 | Idea; another team did it | Video/link |
| 4 | Bench prototype works on the real game piece | Prototype log |
| 5 | Works at speed with realistic driving and abuse | Abuse test, ≥20 trials |
| 6 | Team heritage: our own competition-proven design | Past season data |
| 7 | This exact design on the practice robot | Practice-robot hours |

- [ ] **PDR entry:** every Need mechanism at TRL ≥ 4; TRL 3 → prototype or drop.
- INFERENCE: 1771 skipped prototyping in 2022 because its turret was TRL 6 (5th generation) [S1]. Heritage is a valid reason to skip; confidence is not.
- Concept selection by weighted matrix against requirement IDs, never on looks.

**PDR checklist** (FACT [I2]: preliminary design meets requirements with acceptable risk): - [ ] Block CAD/layout shows packaging, CG, frame perimeter [S20]. - [ ] First weight budget (frc-weight-and-cg-management). - [ ] Top risks listed with FMEA Action Priority (frc-reliability-testing). - [ ] Each subsystem interface has one owner and a frozen envelope.

## Step 3: Detailed design → CDR (before cutting metal)

- [ ] frc-mechanism-design-review on every subsystem.
- [ ] DFM pack and cut list (frc-dfm-manufacturing).
- [ ] Verification plan: every requirement ID maps to a test with a pass criterion.
- [ ] Outside review. FACT [S9]: public review of a CAD-only robot caught a wrist hex at SF ≈ 0.7 and inaccessible electronics. GUESS: ≥1 reviewer from outside the subsystem.
- [ ] Action items (RFA/RID) with owners and due dates. CDR passes with actions; it is not a vibe check.

## Step 4: Integration, test, competition gate

- [ ] Verify every requirement on the practice robot before the event [I13].
- [ ] Run frc-competition-readiness.

## Calendar (FACT [S12] 1690 cadence; adapt)

| Day | Gate |
|---|---|
| 0–1 | SRR |
| 2–5 | Prototypes to TRL 4–5 |
| 6 | PDR; block CAD starts |
| 14 | CDR; interface-driving parts first |
| 15–28 | Practice robot built and wired; weight diet |
| 29–35 | Software and driver week; verification matrix |
| 36–42 | Competition robot; final gate |

## Templates

Review minutes: `Review: PDR | Subsystem: | Date: | Attendees: | Reqs: R-… | Decision: pass / pass-with-actions / fail | Actions: [ID, owner, due]` Decision log: `D-### | date | decision | options | why | requirement IDs | PROPOSED/ACCEPTED/LOCKED`. Never silently edit an ACCEPTED or LOCKED decision; open a new D-### that supersedes it.

## Anti-patterns

CAD before SRR; requirements without verification; 5 functions at 6/10 [S14]; skipping prototypes without TRL evidence; skipping CDR for schedule (INFERENCE: fixes on metal cost far more).

# frc-weight-and-cg-management

**Summary (description):** Use from concept through competition to budget FRC robot mass and center of gravity: per-subsystem allocations with reserve, CAD-vs-scale reconciliation, structured weight diets, and low-CG packaging, before the robot is over weight at inspection.

# FRC weight and CG management

Labels: FACT (source), INFERENCE, GUESS. Source keys in `/workspace/frc-design-kb/top-teams.md`. Tools in `/workspace/frc-design-kb/tools/`: `weight.py` (exit 1 if over) and `cg_tip.py`. Rule limits from `frc-rules-history.md`; never assume this year's limit.

## Why

- FACT [S21]: 1323's practice robot was 130 lb in week 5; by the end they chased 0.01–0.05 lb.
- FACT [S12]: 1690 runs a week-4 diet; found 3 kg in 2025.
- FACT [S19]: 3847's CAD said ~103 lb but the robot was heavier: fasteners and wires weren't in CAD.
- FACT [S14][S13]: 2056: fitting "150 lbs of robot into a 115 lb box."

## 1. Budget at PDR

| Subsystem | Allocation (lb) | CAD est | Measured | Margin | Owner |
|---|---|---|---|---|---|
| Drivebase (+bumpers if counted) | | | | | |
| Electrical + wiring | | | | | |
| Mechanism A | | | | | |
| Fasteners/misc | | | | | |
| **Reserve** | GUESS 8% of limit at PDR, 5% at CDR, 2% at event | | | | |
| - [ ] Check whether bumpers and battery count this year. | | | | | |
| - [ ] Allocations sum to limit − reserve; owners trade with each other, never dip into reserve. | | | | | |
| - [ ] INFERENCE from [S19]: add GUESS 10% on top of CAD for unmodeled fasteners, wires, zip ties, prints until measured. | | | | | |

## 2. Measure early and often

- [ ] Weigh every subsystem at assembly; enter in the CSV.
- [ ] Weigh the practice robot weekly; plot against the limit.
- [ ] If CAD vs scale differs >5% (GUESS), find why and fix CAD materials.

## 3. Diet playbook (FACT [S13] 2056 unless noted, ordered by benefit/risk)

1. Smaller motors where torque allows (X44 instead of X60, ~0.5 lb each).
2. Hollow shafts, thin-wall tube where `beam.py` margin allows.
3. Pocketed plates; formed thin sheet instead of thick plate.
4. Aluminum/nylon bolts in non-structural spots; tapped plates instead of nuts.

5. Parts doing double duty (bumpers or swerve top plates as structure; FACT [S20][S21]).
6. Remove a feature (FACT [S7]: 1002 removed its climber). This is a strategy decision; take it back to frc-systems-engineering.
7. [ ] Every diet change re-runs the margin check in frc-mechanism-design-review. Never pocket a fuse path or fatigue-critical part without recalculating.
8. [ ] Diet log: `item | lb saved | risk | approved by | verified on scale`.

# 4. CG and tipping

- FACT [S1]: 1771 put 11 motors within ~4 in of the floor. FACT [S9]: 1002's CAD robot carried low ballast. FACT [S11]: 254 maximizes weight for traction and base size against tipping. FACT (KB survey): 4414 added low ballast to run a two-high-cone auto.
- [ ] Battery, motors, controllers as low as packaging allows.
- [ ] `cg_tip.py` at max extension: θ = atan(b/h), a_tip = g·b/h; compare with drivetrain μg (measured μ). GUESS gate: no tipping at max drive acceleration with every mechanism extended, otherwise software limits acceleration when extended.
- [ ] Under weight → add ballast low, within the rules.

# 5. Event-day gate

- [ ] Weigh the exact inspection configuration ≥1 week before the event.
- [ ] Margin ≥1 lb (GUESS) for heavier repair parts.
- [ ] Swap-list spares match originals ±0.1 lb (GUESS).

# CSV for weight.py

```
name,qty,lb
swerve module (scale),4,
```

Run `python3 tools/weight.py parts.csv --limit-lb <limit> --reserve-lb <reserve>`. Keep subsystem, source (CAD/scale) and owner in a separate sheet keyed by name.

# industrial-reliability-design-for-frc

**Summary (description):** Use when making an FRC mechanism (or any machine) survive a season the way industrial machines do: derating motors, bearings and printed parts; stress concentration and fatigue at holes and notches; stopping fasteners from loosening; wire and connector mechanical reliability; serviceability and swap time; error-proofing beyond DFM; and pit/event inspection checklists. Complements frc-reliability-testing and frc-mechanism-design-review.

# Industrial reliability design for FRC

Labels: FACT (sourced, keys in /workspace/skill-drafts-frc/SOURCES.md) / INFERENCE / GUESS. Math: `engineering-math-proofs` (Fasteners; Beams; Motors). FMEA and Action Priority: `frc-reliability-testing` §1. Margins of safety and fatigue basics: `frc-mechanism-design-review` §2–3. Poka-yoke basics: `frc-dfm-manufacturing` §5. Onshape GUI only, never FeatureScript.

## 1. Start from a mechanism FMEA, then apply the levers below

- Run the AP FMEA from `frc-reliability-testing` on the mechanism first. Every High-AP row must map to one of §2–§7 here.

## 2. Derating (run parts below their rating, on purpose)

| Item | Rating source | Derate rule | Label |
|---|---|---|---|
| Ball bearings | L10 = (C/P)³ = 90 % survival | For 95 % survival multiply life by a1 = 0.62; 99 % → 0.21 (NHBB/ISO 281 [R2]) | FACT |
| Bearings, life margin | Required service life | Modified life ≥ 2–3× required; ≥ 5× if critical (search synthesis citing SKF [R2b]) | INFERENCE |
| Planetary gearbox | Vendor static rating (MAXPlanetary 4:1 270 N·m, 9:1 180 N·m) | Peak shock torque ≤ 50 % of static rating | GUESS |
| Motors | Stall current | Stator limit set below the traction or stall need; temperature log in practice | INFERENCE |
| Chain | Vendor working load | Size by stiffness and skip first (CD 347386) | FACT/INFERENCE |
| 3D prints (PETG, Onyx) | No public S-N data (FACT gap) | Treat as consumables; ≤ 30 % of the measured break load in service | GUESS |
| - Derating turns most "random" FRC failures into wear items you replace on a schedule (INFERENCE). | | | |

## 3. Stress concentration and fatigue at holes and notches

- FACT [R3]: a circular hole in a wide plate under tension has Kt = 3.0 (Peterson).
- FACT [R4]: Kf = 1 + q(Kt − 1). Notch sensitivity q comes from the Neuber constant and the notch radius.
- FACT [R5] (NIST): for sharp notches, 24S-T4 and 61S-T6 (≈ 6061-T6) aluminum showed notch sensitivity q ≈ 1.
- Rule (INFERENCE): for FRC aluminum near holes, take **Kf = Kt** (q = 1). Do the fatigue check from `frc-mechanism-design-review` §3 with σ_peak = Kt·σ_nominal.
- Design moves that lower Kt (INFERENCE from [R3]):

- Inside corners on plates: fillet ≥ 1/8 in (GUESS); never a sharp pocket corner at a load path.
- Keep holes ≥ 2× diameter from edges and away from the max-moment section of a tube.
- Pocketing: leave a solid ring around bearing and bolt holes (≥ 0.15 in wall, GUESS).
- Polycarbonate: radius every slot end. Bearing holes cracked until reinforced with aluminum (FACT CD 468927).
- Onshape: put the fillet in the sketch or as a Fillet feature right after the pocket so it propagates through configurations. Run the FEA sanity check (`onshape-simulation-fea-sanity`) and read the peak at the hole edge.

## 4. Fastener loosening (vibration and shock)

- FACT [R6]: in the transverse-vibration Junker test (DIN 65151 / ISO 16130), prevailing-torque (nylock) nuts loosen until prevailing torque balances the loosening torque, keeping only residual preload. They are loss-prevention devices, not locking devices, and can fully detach under combined transverse + axial load.
- FACT [R7]: in a Junker test, threadlocked fasteners lost < 1 % of clamp load, vs 40–60 % for unsecured or mechanically locked fasteners after 60 s (Henkel via Design World).
- FACT: a swerve bolt fell out with too little Loctite (CD 472000).
- Rules:
- [ ] Every joint that slips sideways (plates, gussets, sprockets on hubs): proper preload + medium threadlocker (blue). Nylock only where you need quick service and the joint can't slip (INFERENCE from [R6][R7]).
- [ ] Torque stripe (paint pen across bolt and part) on every critical fastener so loosening shows at a glance (INFERENCE; common industrial practice).
- [ ] Set screws: avoid on load paths; if used, threadlock + flat on the shaft (INFERENCE).
- [ ] Write the threadlocker and torque on the drawing (`mech-design-bolted-joints-and-fasteners`).
- [ ] Rivets: pull-test a sample; replace sheared ones with bolts on load paths (FACT: Thrifty elevator rivets sheared).

## 5. Wire and connector mechanical reliability

- FACT [R8]: WAGO 221 lever terminals (as on the PDH): strip 11 mm, bare untinned stranded wire, insert fully, close the lever. Ferrules aren't needed for 221s.
- FACT [R8]: for other clamp terminals use crimped ferrules made with a ratcheting crimper; tug-test every crimp; strain-relieve to a zip tie or carrier, not the terminal.
- FACT: brownouts traced to a loose main-breaker crimp (CD 457396); encoder wires unplugged (CD 472000); a spun motor housing cut about a dozen wires (CD 404942).
- [ ] Service loops at every moving joint, sized at the extreme configuration in Onshape (measure path length in both states).
- [ ] Latching connectors, or a retention clip, on every sensor and CAN joint.
- [ ] No wire runs over a sharp edge without a grommet or edge protector.

## 6. Serviceability (lower mean time to repair)

- FACT [R9]: design for service: make likely failures easy to diagnose, reach, replace, and reassemble; build modules around one function that can be swapped without disturbing healthy parts; use quick-disconnects.
- FACT: 1002 swaps the intake with 2 bolts; the electronics pan comes out with only a drill.
- [ ] Rank parts by expected failures (from FMEA). The top 5 must be reachable without removing another subsystem.
- [ ] One fastener size per module where possible, with tool access in CAD: no blind nuts behind tubes (INFERENCE from [R9]).

- [ ] Swap time gates live in `frc-competition-readiness` §2. Design toward them here.

## 7. Error-proofing beyond DFM (so swaps go back right)

- [ ] Asymmetric bolt patterns or a dowel on any part that can be installed rotated or flipped (FACT: keyed asymmetric interfaces, guide pins [R10]).
- [ ] Left/right mirrored parts get different hole patterns, or are made symmetric so either works.
- [ ] Color-coded or keyed connectors where two plugs could swap (INFERENCE).
- [ ] Label every swap module with name and revision (ties to `onshape-release-and-versioning`).

## 8. Worked mini-example: intake pivot plate, 1/4 in 6061

- Nominal bending stress at the pivot hole ring, from eng-math Beams at the hit load: 50 MPa (GUESS input).
- Peak = Kt·σ = 3.0 × 50 = 150 MPa with q = 1 (FACT [R3][R5]).
- Aluminum endurance ≈ 0.4 × UTS, capped around 19 ksi ≈ 131 MPa, for smooth specimens (FACT via `frc-mechanism-design-review` §3). 6061-T6 UTS 310 MPa → ≈ 124 MPa. 150 > 124, so expect a crack over a season of hits (INFERENCE).
- Fix: add a doubler or boss around the hole, or go to 3/8 in, so nominal ≤ 30 MPa → peak 90 MPa, about 0.73 of endurance before Marin factors. Fillet the pocket corners. Re-run the FEA sanity check.
- Fasteners on the plate: blue threadlocker + torque stripe. The bearing gets a retaining washer (FACT, 1678).

## 9. Inspection checklists

**Before every match (≤ 3 min, GUESS)** - [ ] Tug-test battery leads and main breaker; check torque stripes on the drive and the top-5 FMEA joints. - [ ] Spin each mechanism by hand: no new noise, no play. - [ ] Chain/belt tension by deflection; rope re-tensioned. **End of each day** - [ ] Every crimp and connector on moving joints; tread, grip tape, and roller covers for wear. - [ ] Cracks at holes on plates and polycarb (flashlight + mirror). - [ ] Log findings in the FRACAS sheet (`frc-reliability-testing` §5).

## Gotchas

- Nylocks reused many times lose prevailing torque (INFERENCE).
- Thread-locking a joint with no preload does little. Torque it first (INFERENCE from [R7]).
- Lightening holes near a bolt pattern create Kt hot spots exactly where load enters.

## Hand-off to programmers (numbers only; AI writes no robot code)

- [ ] Current and temperature limits from the derating table (§2).
- [ ] Motors and mechanisms to log in practice (current, temperature, faults) to feed FRACAS.
- [ ] Connectors and sensors flagged in FMEA, so dashboards show when one drops out.
- [ ] Wear-item cycle counts worth logging (climbs, shots, deploys).
- [ ] Swap procedures that change offsets (encoders), so re-zero steps are written down.
- [ ] Write these in the mechanism's spec doc, with units, and update them when hardware changes.

## Done when

Every High-AP FMEA row has a derating, Kt, loosening, wiring, service or error-proofing action; the pit inspection list is printed; and the worked check (Kt·σ < fatigue strength) passes for each load-path plate.

# Simulation

# load-path-simulation-for-assemblies

**Summary (description):** Use when you need to know how much load each part of an assembly actually carries (printed parts next to aluminum, a shaft on bearings vs cantilevered, gussets, bolted joints, multiple supports) before sizing a part, running FEA (Onshape Simulation, FilaSim, FreeCAD/CalculiX), or deciding printed vs aluminum on an FRC robot.

# Load-path simulation for assemblies

> **MANDATORY FIRST STEP:** before choosing a concept or refining any design, run the `design-concept-gate` skill: list only the real constraints, challenge every self-imposed limit (height, part count, look, load case, print orientation), hand-calc 3-6 fundamentally different forms, and pick on numbers. No CAD or sizing until its decision table exists.

Labels: **FACT** ([S#] in /workspace/skill-drafts-print/load-path-simulation-for-assemblies/SOURCES.md), **INFERENCE**, **GUESS**. Onshape is GUI-only, **never FeatureScript**. Related: Onshape Simulation setup/sanity → `onshape-simulation-fea-sanity`; printed-part settings/materials/FilaSim → `3d-print-strength-optimization`; bolts → `mech-design-bolted-joints-and-fasteners`; derates/spares → `industrial-reliability-design-for-frc`; review gate → `frc-mechanism-design-review`; showing the math → `engineering-math-proofs`.

## 1. Core idea: load follows stiffness

- **Series** (one after the other): every part carries the full force; deflections add: $1/k = \Sigma\ 1/k_i$.
- **Parallel** (side by side, forced to move together): forces split by stiffness: $F_i = F \cdot k_i/\Sigma k$; $k = \Sigma k_i$.
- Stiffness formulas (FACT, standard beam theory): axial $k = EA/L$ · cantilever tip $k = 3EI/L^3$ · simply supported centre $k = 48EI/L^3$ · guided-fixed $k = 12EI/L^3$.
- INFERENCE consequence: E(Al 6061) = 68 900 MPa vs PETG HF 1810 MPa (×38). A printed part in parallel with aluminum of similar size carries a few percent; in series it carries 100 %. **Which one is it?** is the first question for every printed part.

## 2. Procedure (hand method, 10 minutes)

1. Draw the free-body diagram of the whole mechanism; find the external load (label GUESS loads, e.g. impact, belt tension, robot weight on a climber).
2. Trace the path from the load to the frame. Mark each part as series or parallel to its neighbours.
3. Count supports vs equilibrium equations: 2 supports on a straight shaft (2D) = **statically determinate** → split from geometry only ($P \cdot b/L$, $P \cdot a/L$). 3+ supports or clamped ends = **indeterminate** → stiffness decides; use the tool or FEA.
4. Compute each part's share, then its stress (axial F/A, bending Mc/I, bearing $F/(d \cdot t)$, shear $F/A$ vs $S/\sqrt{3}$).
5. SF = strength / stress; printed parts need ≥3.33 (≤30 % of break, GUESS rule), metal ≥2 vs yield (GUESS).
6. Only then run FEA on the critical part with **its share** of the load.

## 3. Shaft on two bearings vs cantilevered (most common FRC case)

- Two supports: plates see only radial bearing force; the **shaft** does the bending ($M = PL/4$ at centre).
- Cantilevered: one support must react force **and** moment $M = P \cdot$overhang. A single plate reacts the moment as a couple across its thickness → stress ≈ $F/(d \cdot t) + 6M/(d \cdot t^2)$ (INFERENCE, rigid pin + linear

pressure). Two bearings spaced s apart on a cantilever: reactions ≈ P(overhang+s)/s and P·overhang/s (FACT, statics); close spacing multiplies bearing loads.
- Third support: indeterminate; a far support can be **pulled the opposite way** (uplift). Example (b) below: −3P/32.

## 4. Bolted joints in the path

- Preloaded bolted plates move together → treat as parallel/bonded. Clearance holes without enough preload slip until a bolt bears → the stiffer member takes load only after the gap closes, and a printed part next to aluminum may take everything until then (INFERENCE). Use tight holes/shoulder bolts or rivets where sharing matters.
- Plastic creeps under bolt clamp → preload drops (INFERENCE); use washers, re-torque, metal spacers through prints.
- Preload numbers → `mech-design-bolted-joints-and-fasteners`.

## 5. Contact vs bonded (in any FEA)

- Bonded = glued; transfers tension. Contact = can separate/slide. A bolted printed plate "bonded" to a tube in FEA looks stronger than reality wherever it would actually lift off.
- FACT [S1]: Onshape Simulation is **linear static + modal only**. Connectivity options: *Mates and touching faces* (default; unmated touching faces are **bonded**), *Mates only*, *Bond all touching faces* (artificially stiff). No separating contact (forum, [S2]). Unsupported mates: Tangent, Pin slot, Parallel, Width; relations and composite parts not supported. Fails at ≥100 instances or ≥10 000 contact pairs; aspect ratio >100:1 not recommended. Materials need density, Poisson's ratio, E and yield. Isotropic only → for prints use the **Z** strength as allowable (INFERENCE).

## 6. Tools

1. **`/workspace/loadpath/loadpath.py`** (this box, numpy). `springs` mode for series/parallel networks; `beam` mode for shafts on rigid/spring/clamped supports. Per-part force, share, moment, stress, SF, pass/fail with a printed derate and xy/z strength. Run: `/workspace/cadenv/bin/python /workspace/loadpath/loadpath.py <model.json>`; tests `test_loadpath.py` (26 hand-calc checks pass as of 2026-10-27). Edit `materials.json` with your own break-test numbers. Examples in `/workspace/loadpath/examples/`. Does NOT model bolt-hole slop, buckling, fatigue or stress concentrations.
2. **FilaSim** (filasim.com) for the stress field inside one printed part, with walls/infill and the "Layers" criterion; see `3d-print-strength-optimization`. Feed it the share from step 1.
3. **Onshape Simulation** for metal assemblies or an isotropic sanity check (Professional/Enterprise/Educator plans).
4. **FreeCAD FEM + CalculiX** (free, open source [S3]) when you need contact or nonlinear analysis that Onshape lacks: export STEP, mesh with Gmsh/Netgen, CalculiX solver. INFERENCE: worth it only for one-off critical parts.

## 7. FRC examples (tool outputs; geometry/loads are GUESS scenarios)

**(a) Printed PETG motor mount, 150 N at 60 mm, 50×6 mm section** | config | printed part | other part | verdict | |---|---|---|---| | alone, printed flat | 30 MPa, SF 1.13 | — | FAIL (needs 3.33) | | alone, printed standing (Z) | 30 MPa, SF 0.77 | — | breaks | | bolted to 2x1x1/16 tube behind it (parallel) | 0.046 N (0.03 %) | tube 149.95 N, 2.95 MPa, SF 94 | OK | | clamped flat on tube face (series spacer) | 150 N, 0.075 MPa compression | tube 2.95 MPa | OK |

**(b) Intake roller, 1/2" hex 6061 (I = 1564.5 mm⁴), 450 mm span, 200 N hit, PETG end plates 6 mm, 1.125" bearing** | config | left plate | centre | right plate | shaft | |---|---|---|---|---| | centre hit | 100 N, 0.58 MPa | — | 100 N | M 22 500 N·mm, 105 MPa, SF 2.6, δ 3.5 mm | | quarter-span hit | 150 N | — | 50 N | 79 MPa | | + 1/8" Al centre plate | 81.2 N | 137.7 N | **−18.8 N (uplift)** | 42.8 MPa, δ 0.32 mm | | cantilevered 150 mm, one plate | **176 MPa, SF 0.19 FAIL** | — | — | 141 MPa, SF 1.96 (FAIL vs 2) | Lesson: the hex shaft carries bending between bearings; printed plates are fine. Cantilever = aluminum plate + two spaced bearings.

**(c) Gusset, 2000 N across a tube joint, 40 mm wide, 50 mm between bolt rows** | config | Al 1/8" gusset | printed gusset 5 mm | |---|---|---| | both (parallel) | 1920.5 N (96 %), 15.1 MPa | 79.5 N (4 %), 0.40 MPa | | PETG alone | — | 10 MPa, SF 3.4 (barely OK) | | PAHT-CF alone | — | 10 MPa, SF 9.2 |

# 8. Pitfalls

- Using the total mechanism load on the printed part when it is in parallel with aluminum (over-design), or the share when the aluminum path has slop (under-design). State which assumption you used.
- Treating a 3-support shaft as 2 supports; missing uplift.
- Forgetting stress concentrations (holes ≈ ×2–3, GUESS); the tool reports nominal stress. See Kt rules in `industrial-reliability-design-for-frc` §3.
- Impact: static numbers × a dynamic factor (GUESS ×2–3) for game-piece/robot hits; verify with a break test.

# onshape-simulation-fea-sanity

**Summary (description):** Use when checking whether a bracket, arm, gusset, plate, desk frame or shaft support is strong or stiff enough with Onshape Simulation (linear static or modal), when setting up loads, fixtures, materials and connections in an assembly, when a Simulation result looks wrong or won't run, or when deciding whether to trust FEA versus a hand calculation.

# Onshape Simulation: setup and sanity checks

> **Native features only, never FeatureScript.** Simulation is a built-in assembly panel. No custom features, no Feature Studios. API only lightly (onshape-api-rate-limits).

Labels: FACT = Onshape help (simulation, modal_simulation, asynchronous_simulation) or Learning Center "Introduction to Onshape Simulation" outline; INFERENCE; GUESS. Formulas and the ~20 % agreement rule live in engineering-math-proofs; don't duplicate them here.

## 0. Can I even run it? (FACT)

- Professional, Enterprise and **Educator** plans include Simulation. Free/Student and Standard don't, and Simulation can't run on documents owned by Standard accounts. If the panel is missing, fall back to hand calculations (engineering-math-proofs).
- Analysis types: **linear static** and **modal** only. No thermal, CFD, buckling, nonlinear material, or large-deflection contact (INFERENCE from "linear": results past yield are not meaningful).

## 1. Setup checklist (course order: Materials > Fixtures > Connections > Load > Run > Self-check)

- [ ] **Materials** on every non-fixed instance, including standard-content screws: density, Poisson's ratio, Young's modulus, tensile yield strength (FACT). No Poisson 0.5 or -1; Balsa, Graphite and Silicone rubber don't run (FACT).
- [ ] **Fixture**: at least one fixed instance unless you use **Inertial relief** (FACT). Fix what is really bolted to "ground" (the frame tube), not the part you are studying.
- [ ] **Connections**: pick the method on purpose (FACT):
  - *Mates and touching faces* (default): mates connect, unmated touching faces are **bonded**.
  - *Mates*: only mates connect; other parts can pass through each other.
  - *Bond all touching faces*: ignores mates and stiffens the model artificially. Avoid for real joints. Click **Show connections**: red = not connected (a floating part). Bridge gaps with a **Mate Simulation connection**.
- [ ] Unsupported (FACT): Tangent, Pin slot, Parallel and Width mates; relations (gear/rack); composite parts. Mate limits and mass overrides are ignored. Replace them with Fastened/Revolute for the study.
- [ ] **Loads** (FACT): Force, Moment, Bearing, Pressure, Acceleration, Angular velocity. One instance per load; Bearing and Pressure go on **faces**. Direction = a mate connector's **Z axis**, so place an explicit mate connector pointing the way the load acts.
- [ ] Size limits (FACT): a run fails at >=100 instances or >=10,000 contact pairs; aspect ratios over 100:1 not recommended. Study a **sub-assembly** (bracket + its neighbours), not the whole robot.
- [ ] Long run times out live? Make a **version** and run it **asynchronously** from Results and history (FACT: up to 50 queued per user, 24 h limit, needs a named position unless at the version's default position).

## 2. Reading results (FACT)

- Outputs: von Mises, signed von Mises, **Safety factor = yield / von Mises** (legend 3.5 down to 1.0), max principal stress, displacement. Default units MPa and mm; switch to psi/in if you work in inches.
- The legend max is the **95th-percentile "useful" max per part** because singularities (sharp inside corners, point loads, fixture edges) give fake peaks. Probe the region you care about, not the red pixel at a corner.
- Modal: 5 modes by default, up to 15. Rigid-body modes (about 0 Hz) = the number of open DOF; if you see them in a "fixed" model, something is unconstrained. Keep the first real mode away from excitation (motor/roller rpm / 60 = Hz) (FACT: course; margin GUESS: at least 2x, see frc-mechanism-design-review).
- Multiple simulations give a comparison table; use one study per load case (impact, static hold, worst pose).

## 3. Sanity rules (INFERENCE unless marked)

1. **Hand-calc first** (engineering-math-proofs). Disagree by more than ~20 %? Find out why before trusting either.
2. Check **displacement shape** with exaggeration on: does it bend the way you expect? Wrong shape = wrong fixture or connection.
3. Stress at a **sharp inside corner** keeps rising as you refine: it's a singularity. Add the real fillet or read stress one fillet radius away.
4. Bonded contact (default) is **stiffer and stronger** than a real bolted joint. Bolt preload/slip is not modelled; check bolts by hand (mech-design-bolted-joints-and-fasteners).
5. Aluminium welds: the heat-affected zone is weaker than the T6 value the material card uses (SendCutSend 6061 page). Derate by hand.
6. Polycarbonate and 3D prints: card values are idealised; prints are anisotropic. Treat results as a ranking, not a pass.
7. Impact loads: a static Force of (mass x deceleration) is an estimate. Use the load cases from frc-mechanism-design-review.

## 4. Worked mini-example: 2x1 tube arm, tip load

2x1x0.125 in 6061-T6 tube, 20 in cantilever, 50 lbf tip load, bending about the 2 in depth. - Hand calc: I = (1·2^3 - 0.75·1.75^3)/12 = **0.332 in^4**; M = 50·20 = 1000 lbf·in; sigma = M·c/I = 1000·1/0.332 = **3,015 psi**; delta = F·L^3/(3·E·I) = 50·8000/(3·10.0e6·0.332) = **0.040 in**; SF = 40,000/3,015 = **13.3**. (Sy ≈ 40 ksi from SendCutSend 6061 page; E ≈ 10.0 Msi handbook value, INFERENCE.) - Onshape: assembly with the tube + pivot block; fix the block; Fastened tube-to-block; material Aluminum 6061-T6 on both; Force 50 lbf on the tip end face, direction from a mate connector with Z pointing down; run. - Expect displacement ≈ 0.04 in (slightly more because the block isn't infinitely rigid) and tube-root stress near 3 ksi away from the fixture edge. A 15 ksi spot right at the block edge is a singularity (rule 3). - If the arm also spins at 300 rpm (5 Hz excitation), run Modal and confirm mode 1 is well above 5 Hz.

## 5. Gotchas

- A floating screw (red in Show connections) gives a "rigid body" failure, not a helpful error. Fix it or exclude it.
- Changing a mate to an unsupported type silently drops that connection.
- Assembly configurations are supported; make sure the study is on the configuration you mean.
- Learning Center lessons need sign-in; only the outline and the "Running the simulation study" summary were read. Status icons: warning, gray info, green check. Colour schemes: Viridis (default), Blue Red, Plasma, Rainbow.

## 6. Done when

- [ ] Hand calc and FEA agree, or the difference is explained in the report.
- [ ] Each load case has its own simulation, with fixtures and connections listed.
- [ ] Results quoted away from singularities, with units and the safety factor used.

# Agent tooling

# Cloud agent prompt builder

> **Summary (description):** Use when writing a self-contained prompt for a Cursor cloud coding agent (or any coding agent) on a repo task, especially docs/tools/app work that needs measured results and a reviewable PR.

# Cloud agent prompt builder

A good agent prompt lets a stranger do the job with no follow-up questions and come back with proof.

## Before writing

1. Read the repo's AGENTS.md / README / docs rules (or recall them) - the repo beats any brief.
2. Check memory for the owner's standing rules (model choice, forbidden actions, required logs).
3. Decide if the work belongs to an agent at all (needs the full test suite/build) or is a quick read you can do yourself.

## Prompt sections (in order)

1. **Context**: repo URL, base branch and commit, what exists already (files, functions, prior PRs), and why the work matters.
2. **Goal**: the outcome in one or two sentences, plus measurable targets (numbers, not adjectives).
3. **Exact scope**: files/areas to touch, and what NOT to touch (other agents' areas, prediction math, infra).
4. **Method guards**: measure before changing; for models/backtests: strict time order, held-out set untouched until one final scoring, baselines; for perf: production build, realistic latency.
5. **Constraints**: no merge / deploy / push to main / production migrations / CI triggers; no secrets; no invented numbers; add don't erase (mark DEPRECATED with date); required log entries; branch + one PR.
6. **Acceptance tests / checks**: the exact commands (typecheck, lint, tests, build), known allowed failures, files to revert after build.
7. **Report format**: What / Why / How / Checked / Open, claims labeled FACT / INFERENCE / GUESS, before/after tables, exact test counts, artifact paths.

## After the agent reports

- Read the diff yourself for the core logic; check claims against code and numbers.
- Look for unfair comparisons, measuring the wrong environment, regressions hidden in tables (bundle size, slower rows), and small-sample noise.
- Send it back with specific numbered fixes rather than accepting 'good enough'.
- Never merge without the owner's explicit go-ahead for that PR.