Forge Cost Benefit to Manufacturers when Considering Robotic Automation

(Robotic Weld Example Explained)

August 2025

Introduction

This document explains the numbers shown in Forge Robotics' cost-benefit slide. In a single robotic weld cell, Forge delivers savings at four levels:

- **$32,500 saved upfront** on installation and commissioning.
- **$10,000 saved annually** by eliminating new job programming.
- **$8,000 saved annually** by removing daily re-teaching and corrections.
- **$170,000 unlocked annually** by turning downtime into productive robot hours.

When combined, these add up to a net benefit of **$822,900 over 5 years** after Forge's own costs.

This welding example is used here to illustrate the economics in detail, but the same logic applies across other robotic applications such as palletizing, machine tending, and cutting. The following sections show the breakdown and the assumptions behind each number.

Context

Robotic weld cells are capital-intensive. The hardware (robot, positioner, power source, enclosure) is fixed. The hidden burden lies in **labour**:

- Initial programming and calibration at install *(CLOOS NA, 2025; Lincoln Electric)*.
- New part programming every year *(Visual Components, 2023)*.
- Daily manual re-adjustments when programs drift due to distortion, part variation, and fixture inaccuracies *(Lincoln Electric + integrator interviews)*.

Forge does not change the hardware cost, but it **removes most of these recurring labour costs**. Our pricing: **$30,000 hardware upfront** and **$24,000/year SaaS**. We compare a 5-year baseline vs Forge.

Pricing note

The price of Forge (hardware), vision and compute is subject to change, and will become clearer as further development takes place and production scales. We will update guidance as BOM, compute, and cloud costs are finalized.

Baseline installed cost

Line item	Value	Unit	Source
Robot arm + controller	60,000	USD	(CLOOS NA, 2025)
PLC + HMI + cell controls	20,000	USD	(CLOOS NA, 2025)
Robotic power source + torch	20,000	USD	(Lincoln Electric)
Positioner (1–2 axis)	40,000	USD	(Lincoln Electric)
Safety enclosure + fencing	9,000	USD	(Vention, 2024)
Fixtures and tooling	35,000	USD	(ResearchGate, 2014)
Hardware subtotal	**184,000**	USD	
Engineering design + build	30,000	USD	(CLOOS NA, 2025)
Install + wiring + calibration	15,000	USD	(CLOOS NA, 2025)
Initial programming + commissioning	20,000	USD	(Visual Components, 2023)
Services subtotal	**65,000**	USD	Labour
Total installed cost (baseline)	**249,000**	USD	Matches industry ranges

1. Install labour (one-time)

Formula

$$\text{Install savings} = \text{Install labour cost} \times \text{Percent reduced}$$

Numbers

$$\text{Install savings} = 65{,}000 \times 50\% = 32{,}500$$

Answer: Forge saves about **$32,500** at install. (Source: CLOOS NA, 2025)

2. New job programming (annual)

Factories spend significant time programming new parts, pendant programming can take weeks. OLP reduces this by 50–80% *(Visual Components, 2023)*. Here we assume 12 simple jobs/year (12h each) and 2 complex jobs/year (80h each). Rate = $50/h.

Formula

$$\text{Programming cost} = (\text{Simple jobs} \times \text{Hours simple}) + (\text{Complex jobs} \times \text{Hours complex}) \times \text{Rate}$$

Numbers

$$\text{Programming cost} = (12 \times 12 + 2 \times 80) \times 50 = 11{,}200$$

Answer: Annual programming cost = **$11,200**.

Forge reduces this by 90%:

$$\text{Programming savings} = 11{,}200 \times 90\% = 10{,}080 \text{ per year}$$

3. Daily re-adjustments (annual)

With "dumb" systems, operators manually correct programs daily due to heat distortion, part deviation, and fixture inaccuracies *(Lincoln Electric, industry interviews)*. We assume 1 h/day lost, 250 days/year, at $40/h.

Formula

$$\text{Adjustment cost} = \text{Hours per day} \times \text{Days per year} \times \text{Rate}$$

Numbers

$$\text{Adjustment cost} = 1 \times 250 \times 40 = 10{,}000$$

Answer: Annual adjustment cost = **$10,000**.

Forge removes 80% of this overhead:

$$\text{Adjustment savings} = 10{,}000 \times 80\% = 8{,}000 \text{ per year}$$

4. Contribution margin per cell-hour

We define contribution margin per cell-hour as revenue contribution minus variable costs.

Formula

$$\text{Contribution/hour} = (\text{Parts/hour} \times \text{Contribution/part}) - (\text{Consumables/hour} + \text{Operator fraction} \times \text{Rate})$$

Numbers

$$\text{Contribution/hour} = (10 \times 15) - (13 + 0.2 \times 40) = 150 - 21 = 129$$

Answer: $129/hour, use $100/hour as conservative baseline.

5. Downtime & hours/day uplift

Small job shops typically plan around a single 8-hour shift *(just-plan-it SMB example, 2020)*. Manual welding arc-on time is often 10–12% *(Miller, 2020; AWS Welding Digest, 2024)*. Optimized robotic cells with dual-station setups report 60–75% arc-on time *(FSMD, 2022; Novarc, 2025)*. Forge's real-time adaptability allows reliable unmanned running for a second shift, making **16 production hours/day** realistic for small shops *(Okuma, 2021; FANUC case study)*.

Why Forge enables higher uptime

- **Traditional barrier: on-robot programming stops production.** In high-mix shops, every new job requires downtime at the pendant, often hours or days. Forge eliminates this with AI-driven programming directly from scans, keeping the cell productive. *(Visual Components, 2023)*
- **Traditional barrier: daily drift corrections.** Heat distortion, fixture inaccuracies, and inconsistent parts force manual touch-ups, which eat 1–2 hours per day. Forge's vision-guided real-time correction removes most of these stops, keeping the robot running. *(Lincoln Electric interviews)*
- **Traditional barrier: limited to one attended shift.** Most small shops don't risk running overnight because they cannot trust the robot to adapt if parts deviate. Forge's adaptability allows reliable second-shift or lights-out operation, proven in analogous machine-tool cases. *(Okuma, 2021; FANUC case study)*
- **Competitor systems make partial gains.** Some vision systems can handle seam tracking or minor part variation, giving incremental uptime improvements. But they remain rule-based and rigid, they cannot fully reprogram, adapt across different part geometries, or self-correct in real time. Forge's AI-native architecture unifies scanning, programming, and adaptive execution, so it captures *all three levers at once*: no pendant downtime, no daily touch-ups, and safe lights-out running.
- **Net effect:** Baseline utilization is typically ∼35% of an 8h shift (2.8h arc-on/day). With Forge, the same cell can run 16h/day at 60% arc-on (9.6h/day). That is a 3.4× increase in productive hours, translating directly into more throughput and margin.

Worked calculation

Formula

$$\text{Uptime value/year} = (\text{Forge arc-on hours/year} - \text{Baseline arc-on hours/year}) \times \text{Contribution/hour}$$

Numbers

$$\text{Uptime value/year} = (2{,}400 - 700) \times 100 = 170{,}000$$

Answer: $170,000 per year.

Formula

$$\text{Uptime value/year} = (\text{Forge arc-on hours/year} - \text{Baseline arc-on hours/year}) \times \text{Contribution/hour}$$

Numbers

$$\text{Uptime value/year} = (2{,}400 - 700) \times 100 = 170{,}000$$

Answer: $170,000 per year.

Conservative case (worst case with Forge)

Even if Forge only enabled 12h/day at 50% arc-on (6h/day productive time), compared to the baseline of 2.8h/day, the cell still delivers about 900 extra arc-on hours per year.

Formula

$$\textit{Uptime value/year (conservative)} = (\textit{Forge arc-on hours/year} - \textit{Baseline arc-on hours/year}) \times \textit{Contribution/hour}$$

Numbers

$$\textit{Uptime value/year (conservative)} = (1{,}500 - 700) \times 100 = 80{,}000$$

Answer: *Even in the **worst case**, Forge adds about $80,000 per year. This floor case still delivers more value than any competitor system, because no other vision layer simultaneously removes pendant downtime, daily corrections, and unlocks a second shift.*

6. Annual savings summary

Category	Savings (USD)
Install labour (one-time)	32,500
New job programming (per year)	10,080
Daily re-adjustments (per year)	8,000
Uptime value (per year)	170,000
Annual recurring savings	**188,080**

7. Forge costs

Item	Cost (USD)
Hardware (year 1)	30,000
Software (per year)	24,000

8. ROI over 5 years

Year 1: In the first year, Forge saves both the one-time install labour reduction and the annual recurring savings, but we must subtract the cost of Forge hardware and software.

$$\text{Net Year 1} = (\text{Install savings} + \text{Annual savings}) - (\text{Hardware} + \text{Software})$$

$$\text{Net Year 1} = (32{,}500 + 188{,}080) - (30{,}000 + 24{,}000) = 166{,}580$$

Answer: Forge delivers a net benefit of **$166,580 in Year 1**.

—

Years 2–5: In following years, the one-time install saving no longer applies. The savings come only from recurring annual savings, minus the Forge software subscription.

$$\text{Net Year t} = \text{Annual savings} - \text{Software}$$

$$\text{Net Year t} = 188{,}080 - 24{,}000 = 164{,}080$$

Answer: Forge delivers **$164,080 per year** in net benefit during Years 2–5.

—

Cumulative 5-year net: To calculate the total impact, we add Year 1's benefit to four more years of recurring net benefit.

$$\text{Cumulative 5-year} = \text{Net Year 1} + 4 \times \text{Net Year t}$$

$$\text{Cumulative 5-year} = 166{,}580 + 4 \times 164{,}080 = 822{,}900$$

Answer: Over 5 years, Forge delivers a total net benefit of **$822,900**.

Interpretation

- Forge removes recurring install labour, new-job programming, daily re-adjustments.
- Forge enables **1,700 extra arc-on hours/year**, valued at **$170,000/year**.
- Net 5-year ROI: **$822,900** positive.

9. Extension to Other Industrial Robotics

The same economics that Forge delivers in welding; reducing downtime, programming overhead, and unlocking extra uptime, apply directly to other categories of industrial robotics.

Examples

- **Palletizing & material handling.** Traditional systems require extensive calibration and teach points whenever box sizes or SKUs change. Studies show up to 20–30% of engineering time is lost on reconfiguration. *(RIA, 2022; ABB Logistics, 2023)*
- **Machine tending.** Small machine shops using robot loaders often fail to run overnight because the robot cannot adapt to part variations or fixture slippage. Lights-out CNC case studies show 2× throughput when confidence is achieved. *(Okuma, 2021; FANUC case)*
- **Cutting, grinding, surface prep.** Offline programming dominates cost in high-mix cases, often 50–70% of deployment time. *(Visual Components, 2023)*

Worked examples

Palletizing downtime

$$\text{Annual lost hours} = \text{Changeovers per year} \times \text{Hours per changeover}$$

$$\text{Annual lost hours} = 200 \times 2 = 400$$

If each hour = $80 contribution, this is **$32,000/year** saved with adaptive programming.

—

Machine tending (lights-out unlock)

$$\text{Extra hours/year} = (16 - 8) \times 250 = 2{,}000$$

At \$100/hour contribution, this is **\$200,000/year** incremental value.

—

Cutting OLP reduction

$$\text{Programming cost} = 400 \times 50 = 20{,}000$$

Forge's auto-programming reduces by 80% → **\$16,000/year** saved.

—

Conclusion

Whether welding, palletizing, tending, or cutting, the formula repeats:

$$\text{Net value} = (\text{Extra uptime} + \text{Reduced programming}) \times \text{Contribution/hour}$$

Forge's AI-native approach applies across the board, compounding cost savings in any robotic deployment where downtime and variation are the limiting factors.

Sources

Source	**Details**
CLOOS NA (2025)	Robotic weld cell cost ranges, pre-engineered vs custom. `https://www.cloosna.com/how-much-does-a-robotic-welding-cell-cost-in-2025/`
Lincoln Electric	Power Wave R450 pricing, ANNEX positioners catalogue. `https://www.lincolnelectric.com`
Visual Components (2023)	OLP guide, AFRIT case study. `https://www.visualcomponents.com/blog/offline-robot-programming-olp-the-complete-guide-with-examples/`
Industry interviews	Lincoln Electric + integrator interviews on daily manual corrections. `https://www.lincolnelectric.com`
Vention (2024)	Robot cell enclosures configurator. `https://vention.io`
ResearchGate (2014)	Fixture design review: fixtures often 10–20% of system cost. `https://www.researchgate.net/publication/263164980_A_review_of_fixture_design_for_flexible_assembly`
just-plan-it SMB (2020)	Job shop capacity, one shift, 8h/day. `https://www.just-plan-it.com/smb-production-scheduling-blog/why-job-shop-scheduling-starts-with-defining-resource-capacity-video`
Miller Electric (2020)	Average welding arc-on 10%. `https://www.millerwelds.com/resources/article-library/what-is-welding-arc-on-time-and-are-you-overestimating-it`

Source	**Details (cont.)**
AWS Welding Digest (2024)	Operator factor 10–12%. https://www.aws.org/magazines-and-media/welding-digest/wd-feb-24-weld-setting-optimization-for-collaborative-robots/
FSMD (2022)	Dual-station robotic cells arc-on 75%. https://fsmdirect.com/empowering-automation/
Novarc (2025)	Pipe spool welding arc-on 60–80%. https://www.novarctech.com/resources/blog/spool-welding-robot/why-the-spool-welding-robot-is-the-best-investment-for-pipe-fabrication-operations/
Verbotics TRT (2022)	HMLV case, arc time 40–75%. https://verbotics.com/news/2022-08-02-trt-case-study/
Okuma (2021)	Lights-out automation, overnight. https://www.okuma.com/blog/dont-be-afraid-of-the-dark-running-24-7-lights-out-manufacturing
FANUC case	Prototype shop lights-out overnight. https://www.fanucamerica.com/case-studies/kad-prototype-shop-lights-out-manufacturing-with-automation
PracticalMachinist	Consumables cost $13/arc-hour. https://www.practicalmachinist.com/forum/threads/order-of-magnitude-for-operational-cost-of-robotic-weld-cell.355535/
RIA (2022)	Robotics Industry Association logistics automation survey: calibration/changeovers consume 20–30% of engineering time. https://www.automate.org/news
ABB Logistics (2023)	ABB case study: robotic palletizing ROI in logistics operations. https://new.abb.com/products/robotics/industries/logistics
Okuma (2021, CNC tending)	Case for lights-out CNC machining, doubling throughput once reliable. https://www.okuma.com/blog/dont-be-afraid-of-the-dark-running-24-7-lights-out-manufacturing
FANUC CNC tending	Small-shop FANUC tending robots run unmanned overnight once debugged. https://www.fanucamerica.com/case-studies
Visual Components (2023, cutting)	Offline programming in cutting/grinding adds 50–70% of project cost. https://www.visualcomponents.com/blog/offline-robot-programming-olp-the-complete-guide-with-examples/