VOLUME II · ESSAY 03

When the Engineering Loop Closes

From simulation to experiment to action.

Christophe Pennetier, André-Benoît De Jaegere

Generative Engineering · Volume II · written from October 2026, two years after Volume I · https://generative-engineering.aquia.studio/volume-ii/closing-the-loop/

CONTENTS

A mirror, not a twin 4
A network that needs less clairvoyance 6
When another simulation costs almost nothing 10
The laboratory answers back 12
Rehearsal meets matter 15
The people inside the model 17
The organization is part of the experiment 18
The right to stop the loop 21
Sources 24

Before he becomes Captain America, Steve Rogers has a problem with altitude.

Halfway through a training run at Camp Lehigh, a flag hangs at the top of a pole. The drill sergeant tells the recruits that the first man to bring it down rides back to camp with Agent Carter, in her jeep; nobody has managed it in seventeen years. The recruits are considerably better equipped for the assignment than Rogers: bigger, stronger, and apparently persuaded that the relationship between a man and a flagpole is fundamentally a test of biceps. They climb. They slide back down. Rogers walks to the base, pulls out the pin that holds the pole upright, and lets it fall. He picks the flag off the grass, hands it to the sergeant and climbs into the jeep. No serum required.[1]

The scene lasts a minute. I have thought about it for rather longer, and I am not the only one: it already circulates as a lesson in thinking outside the box, in classroom clips and leadership posts, and the lesson usually ends when the flag reaches the ground.[2][3] This essay is about what happens after.

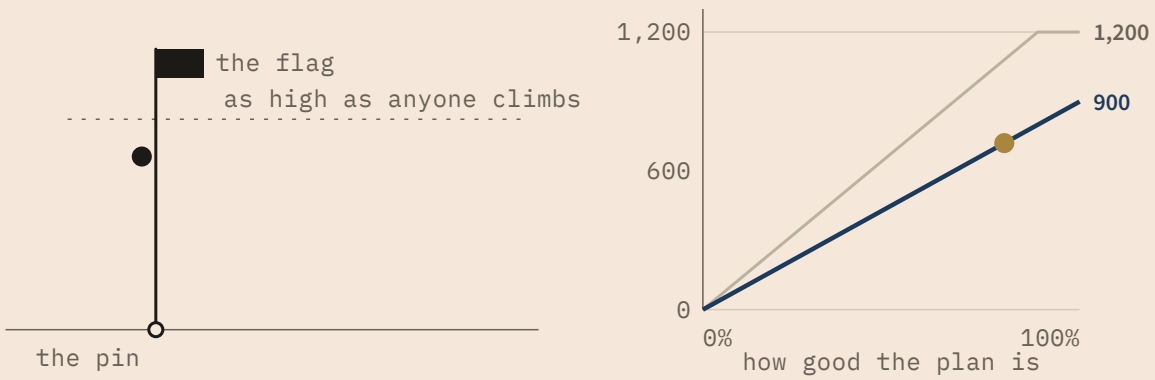
I should confess that, by training, I am a climber. From a young engineer's years at JPL to a doctorate and the years of research that followed, my

mathematics has mostly consisted of getting up poles other people had planted. Optimization people love a tall pole. It justifies the optimizer.

Everyone in the scene is trying to improve the climber. Rogers changes the arrangement that made climbing necessary. The distinction sounds obvious once the flag is on the ground. Before that, it is remarkably easy to miss, and organizations can devote years to becoming better climbers.

I meet the same problem in logistics. A network has its routes, departure times, service promises, vehicle restrictions, staffing patterns and rules about which parcel may travel where. Demand rises. A hub fills. More parcels miss their connections, and we ask the optimizer to perform a small miracle inside a configuration whose assumptions have just become impossible to satisfy. Sometimes a better algorithm helps enormously. Sometimes the consequential decision was available a month earlier: change a connection, reserve a different class of vehicle, move a shift, revise a priority rule, or open a route the original problem did not allow. By the time we ask how to allocate the remaining seats, we have already decided most of what can happen (Fig. 1).

FIG. 1 **Even a perfect plan leaves 300 of 1,200 parcels behind when the truck leaves at 21:30; moving the departure to 21:45 brings every parcel within reach, and the plan still decides how many make it.**



On the page, move how good the plan is, or how hard the recruit climbs, and choose the arrangement; in print, a plan 80 percent as good as it can be, as given, with the later departure in grey. An illustrative depot: two late trucks bring 1,200 parcels at 21:00 and the loading team can sort at most 30 a minute. The plan's quality is an illustrative assumption: the share of that top rate the plan achieves, so 80 percent means 24 parcels a minute. As given, the truck leaves at 21:30; pulling the pin moves the departure to 21:45. The pin is not free: a later departure arrives later downstream, a cost this figure leaves out. The pole is a drawing, not the film's set, and its proportions are illustrative.

Built

This is no argument against optimization. An optimizer can search over network designs, schedules, policies, even over formulations of the problem, when we give it those variables. It does not have to imitate a muscular recruit. But someone, human or machine, has to recognize that the pin belongs in the decision space. An excellent solution to the wrong formulation remains an excellent solution to the wrong formulation.

None of this is new. Pólya told his students to restate the problem; Herbert Simon held that solving a problem simply means representing it so that the solution becomes transparent; Russell Ackoff distinguished solving a problem from dissolving it, by redesigning the system that has it.[4][5][6] Machines have searched over the form of a model before: in 2009 Michael Schmidt and Hod Lipson's program recovered conservation laws from the motion of oscillators and double pendulums, with no physics given in advance.[7] What is changing is how broadly we can search those representations, and how closely that search can be connected to simulation, experiment and revision.

Nor does changing a configuration abolish computational complexity. Many of the problems behind a logistics network are NP-hard: no one knows a method that stays fast on every instance as the instances grow. A redesigned instance may become easier, more resilient or simply more useful without making the whole family easy. The practical claim is more modest and more interesting: we can sometimes spend intelligence upstream so that less intelligence, less information or less frantic intervention is needed downstream.

And Rogers does get the serum, a little later. We can have the better formulation and the stronger solver.

That possibility is why this third essay matters to Generative Engineering. A system that can generate a design, simulate its consequences, propose an experiment, observe what actually happens and revise its next proposal changes the work of engineering. We are beginning to assemble such systems, and the vocabulary has moved faster than the evidence. So the question is how much of the apparent loop is real: which parts are measured and which are assumed, which have been demonstrated, and which survive outside the demonstration. The promise is a shorter journey from imagination to evidence. The hard part is letting the evidence change the imagination.

A mirror, not a twin

The first volume of this study already treated digital twins as active systems, models of a real asset that are kept up to date with its data and used to predict and decide.[8] It would be convenient to invent a past in which everyone merely admired dashboards, but that was never the ambition. In “The Living Twin” I argued that most digital twins are not twins but portraits, reassuring and unable to argue back; that a useful one is closer to a wind tunnel, built to let bad ideas fail safely and early; and that a twin becomes living only when it closes the loop between recommendation and reality.[9] This essay is about what closing that loop costs.

Working with simulation, I have grown uncomfortable with the word twin. It invites a peculiar competition: how much of reality can we reproduce before anyone asks what the reproduction is for? I prefer to speak of a digital mirror. The word acknowledges a relationship to something real, a need for calibration and an angle of view, and it leaves room to change what we see. A model of a factory can become a factory with different stations; a representation of a transport network can become a network that does not exist yet. That departure from resemblance may be the mirror’s most valuable feature.

Consider an imaginary parcel depot. Its simulation reproduces the average processing time almost perfectly, and the average is reassuring. Unfortunately, the evening failure happens when two late trucks arrive together, the same loading team must handle both, and a connecting departure cannot wait. A model can match the daily average and miss the coincidence that causes the loss. Now reverse the priorities: represent the queues, the shared staff, the cut-off times and the uncertainty about arrivals carefully, and leave the colour of the loading doors unspecified. This model looks less like the depot and answers much better the question of whether another overlapping shift is worth paying for. Fidelity is a budget; the decision says where to spend it.

A model of the training camp accurate to the last rivet would not have suggested pulling the pin either. That took a question about the pole, not a better picture of it.

I have not always used the word mirror kindly. The first volume called twins “living, learning mirrors”; in my notes for this essay, the mirror was the passive thing a twin should outgrow; and “The Living Twin” kept the

twin for the good version and called the bad one a portrait. Consistency of vocabulary is not among my virtues. History, at least, is on the mirror's side: before NASA's John Vickers suggested the name digital twin, Michael Grieves called his model the mirrored spaces model, then the information mirroring model.[10] Engineers who study validation use the word too, in a technical sense of their own.[11] And the best-known parable of reframing in management already contains one: tenants complain that a lift is slow, and instead of faster motors someone puts mirrors beside it, so that people stop noticing the wait.[12] My mirror is meant for the opposite purpose. It should make us notice.

This is a preference about emphasis, not a discovery that the literature got twins wrong. The National Academies' 2024 report defines a digital twin by what it does: a virtual representation, updated from its physical counterpart, with a predictive capability, that informs decisions which realize value. It asks for a representation fit for its purpose, whether high-fidelity, simplified or surrogate, and it calls the two-way interaction between the virtual and the physical central to the idea.[13] A mature twin already does much of what I want from a mirror. The trouble is the expectation the word creates in a meeting, particularly once somebody has paid for an impressive three-dimensional model.

Calibration helps: adjusting a model's parameters until its outputs agree with observed measurements. But agreement with the observations used to tune a model cannot by itself establish that the model will behave correctly in a new situation. Marc Kennedy and Anthony O'Hagan's foundational work on calibrating computer models separates two things the word mixes: uncertain parameters, which data can tune, and the discrepancy between the model and the physical system, which tuning can hide.[14] A well-tuned model can still be wrong in a way that matters. NASA's standard for models and simulations makes the discipline explicit: credibility is judged for an intended use, from verification, validation, uncertainty and the evidence behind that particular application.[15] A model can be good enough for one decision and inadequate for another, even when both concern the same machine.

So the demanding question is not whether the digital representation is accurate. It is: accurate enough to choose what, under which conditions, at what cost of being wrong?

There is a mathematical tradition behind this shift. In "Smart 'Predict, then Optimize'", Adam Elmachtoub and Paul Grigas train prediction models with the downstream decision in view: they measure a forecast's errors by

what those errors cost in the optimization that uses it, rather than assuming a generally better forecast makes a better decision.[16] Their guarantees belong to their assumptions; the lesson travels further. Two errors of the same size can have entirely different consequences. Suppose the choice is whether to add a shift. A forecast that moves from ten thousand parcels to ten thousand and ten may change nothing; a smaller error near the threshold at which the depot overloads may change everything. An engineer should want to know which uncertainty can reverse the decision, not which column of predictions has the most attractive average score.

This is also where simulation becomes dangerously persuasive. We change a parameter, the simulated result changes, and the animation appears to demonstrate a cause. It has demonstrated a consequence inside the model. Whether the same intervention has the same effect in the world depends on whether the model's mechanisms and assumptions hold there; causal inference formalizes exactly this dependence on data and assumptions, and a counterfactual-looking screen does not deliver it automatically.[17] A mirror is therefore an instrument with a declared purpose. We should know what it measures, where it is uncertain, and which observation would make us stop believing it. Without those, calibration can become an elaborate way of teaching a model to agree with us.

A network that needs less clairvoyance

A parcel seems simple when it travels alone. Give it a destination, a few possible routes and their prices, and find a suitable path. Now add a second parcel. If there is unlimited room everywhere, it can be treated much like the first. Add limited aircraft capacity, finite loading crews, a closing gate and several carriers protecting their own commitments, and each decision changes what remains possible for the others. Capacity creates a conversation between decisions. It is also an excellent way to discover which data nobody is willing to share.

In the kind of network I have in mind, the operator does not own every vehicle, warehouse or booking system. A carrier may expose a price and an acceptance decision without exposing the true capacity left behind them. A simulator can contain an exquisitely detailed capacity field that the deployed policy will never receive, and declaring the missing column mandatory does not make it arrive. One response is to pursue better

visibility, and sometimes that is exactly right. Another is to ask whether the network can be designed to depend less on the visibility we cannot obtain.

That is the hypothesis I want to explore with a digital mirror. Create a family of plausible worlds: different demand patterns, correlated disruptions, hidden capacity limits, delayed information, imperfect compliance. In those worlds, redesign the network and the policies around it. Then test the resulting design with a controller that sees only what will actually be observable in operation. The distinction matters. The simulator may need the hidden capacities to generate plausible outcomes; the controller must not quietly borrow them to make its decisions. Otherwise we have tested a clairvoyant dispatcher and hired an ordinary one.

The aim is a configuration that performs acceptably across a range of conditions with modest information. A cross-connection may make rerouting possible; a reserved option may soften a capacity shock; a different division of priority classes may keep urgent freight from being trapped behind freight that could have waited. None of these makes the missing information exist. They may reduce how much damage its absence can do. If this essay had to fit on the back of a delivery note, it would read: design so that what we cannot know matters less, and give experience a real way to change the design.

The idea has a beautiful ancestor in William Jordan and Stephen Graves's work on manufacturing flexibility, published in 1995. They showed that a carefully connected pattern of limited flexibility, each plant able to make a few products, linked in a long chain, captures most of the benefit of letting every plant make every product. The structure of the connections mattered, not merely their number.[18] Their setting does not prove the logistics hypothesis above; it shows why architecture deserves a place beside intelligence in explaining performance. A second ancestor is robust decision making: Robert Lempert, Steven Popper and Steven Bankes argued for examining many plausible futures and looking for strategies that remain useful across them, rather than making one preferred forecast carry the whole decision.[19] A large collection of simulated worlds is not automatically a probability distribution. Sometimes it is a disciplined collection of ways to be wrong.

The practical experiment has to keep that discipline. Compare the current design and the proposed one under the same conditions, and give their deployed policies the same information. Count the cost of creating flexibility, the extra computation and the disruption of the change. Hold out combinations of shocks during development, and test cases in which

several apparently independent alternatives fail together: a network with three routes through the same vulnerable hub does not have the diversity its diagram suggests. An all-seeing policy is useful as an upper bound, since it tells us what perfect information would be worth, but it cannot be the only baseline. If the redesign wins only when its controller sees hidden capacities, the experiment has found the value of information, not the value of the redesign.

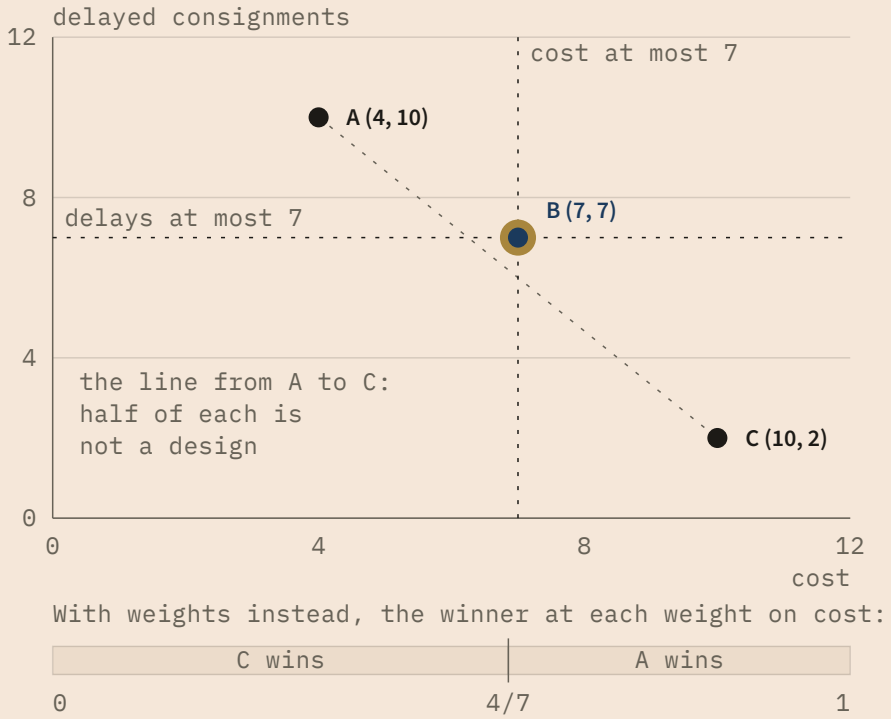
There is no general guarantee that a topology will absorb uncertainty. Some bottlenecks are physical; some information is indispensable. The claim has to be made testable, and the test allowed to reject it. In operation, the mirror must be recalibrated and challenged by observed queues, refusals, delays and costs. And the degrees of freedom have different clocks: a route changes faster than an airport, and a shift faster than a terminal.

This is close to what I am pursuing in my own research on decision quality, where one page is headed, literally, “Pull the pin”: make one small commitment, let the same solver finish, compare the result with the untouched baseline, and keep the change only if it pays for itself.[20] A strong solver need not be replaced to benefit from intelligence around it. In one notebook experiment, a thin wrapper committed a few high-value units, passed the rest of the problem to an unchanged planner, checked the result and kept the baseline whenever the candidate was worse. It gained on thirty fresh test worlds and never returned a worse scored solution than the baseline; the first accounting, however, omitted the extra computation.[21] The detail I value most came afterwards. Two later analyses overturned my explanation of why it helped: the gain came from forcing part of a route, not from the mechanism I had first described.[22] [23] The number stood; the story did not. Even a successful result does not entitle its author to the first convenient mechanism.

There is a further complication. Better according to which preference?

Imagine three feasible network designs, scored on cost and on delayed consignments, both of which we want small. A costs 4 and delays 10; B costs 7 and delays 7; C costs 10 and delays 2. None is simply worse than another: A is the cheapest, C the most reliable, B the balanced one. Now give each quantity a weight and add them up, the most common way of turning two goals into one. Whatever weights you choose, A or C wins; B never does (Fig. 2). Ask instead for a cost of at most 7 and at most 7 delays, and B is the only acceptable design.

FIG. 2 **No weighting of cost against delays ever chooses the balanced design B; two plain requirements choose B alone.**



On the page, move the weight on cost, or switch to two requirements; in print, the requirements, with the winner at every weight beneath. Three illustrative designs; lower is better on both axes. A weighted sum chooses the design that a line of equal score, tilted by the weight, touches first: A or C, never B, which lies above the line from A to C. B beats A only when the weight on cost is below one half, and C only when it is above 0.625. The line from A to C is not a set of designs: half of one network and half of another is not a network.

Built

This is why the Pareto frontier, the set of designs for which no feasible alternative is at least as good on every criterion and strictly better on one, matters more than a smoothly drawn compromise. Where that frontier is not convex, where a design that nothing beats lies above the line joining two others, weighted sums cannot reach it; Indraneel Das and John Dennis gave the geometry in 1997.[24] The line between A and C looks like a menu of compromises, but it describes mixtures that cannot be built as one operating network. More careful methods handle this; a convenient slider between cost and service does not, by itself.

Simulation can expose these choices and optimization can search them. Neither can decide, without an expressed preference, how much a missed medical shipment should count against another vehicle, or whether a cheap network is acceptable if it fails badly under a plausible disruption. The engineer’s growing power is to generate and test configurations before

they become commitments. The responsibility that comes with it is to keep the choices visible while doing so.

When another simulation costs almost nothing

For a long time, the price of asking another question limited the questions an engineer could ask. If every candidate needs a difficult mesh, a long computation and an expert to interpret the result, exploration becomes selective. That selectivity can be intelligent. It can also keep a team near the designs it already understands. A large gain in simulation speed is a chance to widen the search, revisit assumptions and try combinations that previously seemed too expensive to examine.

Neural operators are one important route. Instead of learning a mapping between fixed lists of numbers, they learn mappings between functions: from a description of a physical field and its conditions to another field, such as how pressure, velocity or temperature evolves. A perspective by Kamyar Azizzadenesheli, Anima Anandkumar and colleagues in Nature Reviews Physics describes their use in scientific simulation and in inverse design, with speedups the authors put at four to five orders of magnitude in some applications.[25] They are learned approximations to families of problems, and each family is a domain in which their accuracy must be established. The distinction between forward and inverse work matters. Forward, we ask what a proposed geometry does; inverse, which geometry would produce a desired behaviour. Cheap, differentiable approximations make it practical to move between those questions repeatedly, so the candidate becomes something we adjust in response to the predicted physics rather than something we submit once and wait to hear about.

The Well, released in late 2024, supplies about 15 terabytes across sixteen physical datasets for developing and comparing such models.[26] Infrastructure of this kind is easy to overlook beside an attractive demonstration. A model that handles one favourite equation is a different scientific proposition from one evaluated across a varied collection of systems that others share. Shared tests make a claim harder to decorate.

By 2026 the commercial ambition had become explicit. Accelerated Understanding, founded by Anima Anandkumar and Benedikt Jenik, describes “universal physical AI models” with a full representation of three-dimensional space and time, trained to obey known physical laws

because, as it says, a result can be checked against its governing equation. World models built for video, it argues, settle for plausible appearance; engineering cannot.[27] The company goes further than exploring designs: it says it is replacing experiments with AI that can simulate and understand physics, while conceding that the results can still be checked in the laboratory. Keep that concession in mind. This essay is mostly about it.

These are a company's claims, with self-reported demonstrations, not independent proof of a universal physics engine. The money says something else, and something real. PhysicsX raised $300 million in June 2026, at a valuation of about $2.4 billion, in a round led by Temasek, with Siemens and NVIDIA among its existing investors.[28] A financing round measures investors' commitment, not a surrogate's accuracy or a customer's return.

More revealing is the company's own candour a few weeks later. It stresses the cost of generating simulations, curating data and keeping models current, and notes that where accelerated conventional solvers already run fast, the economics of a dedicated AI model become "far less obvious": justified for many applications, it adds, but not all. The essay then argues that its foundation models cut those costs, which is the pitch, and fair enough.[29]

An instant answer is less impressive if preparing the question takes six months. Equally, six months may be an excellent investment if they serve thousands of consequential decisions over several product generations. The whole experiment has to be counted: data preparation, training, integration, computation, physical validation, rework and maintenance. The denominator is not always seconds per simulation. It may be days per accepted design, or cost per improvement confirmed by experiment.

World models widen the discussion without making all simulation interchangeable. Google DeepMind's Genie 3, announced in August 2025, generates interactive environments in real time that stay visually consistent for a few minutes; its own announcement lists limits on the actions an agent can take, on interactions between agents, on geographic accuracy and on duration.[30] Such a model may provide useful worlds for training agents. A convincing film of a bridge is not yet evidence about its load-bearing capacity. Nor is the line simply between neural networks and physics: learned components can serve an engineering simulator, and classical solvers predict badly when their inputs or material assumptions are wrong. What matters is which quantities and interventions have been tested. For a robot, contact and friction may matter more than a beautiful

wall; for a cooling system, the unphotogenic temperature field may be the whole point.

The industrial evidence becomes useful when it names a workflow. BMW reported in June 2025 that its planners were working in the digital twins of more than thirty production sites, with automated checks for collisions before new models are launched: what once took almost four weeks of real testing now takes three days of simulation. The often-repeated figure of up to 30 percent lower planning costs was a projection, not a measurement.[31] The concrete change is easy to appreciate: a collision found while the installation is still editable, rather than when the line must stop to reveal it. The saved time matters because of where it falls in the chain of commitments. In January 2026, Siemens reported that its new digital-twin software, used at select PepsiCo facilities in the United States, had delivered a 20 percent increase in throughput on initial deployment: [32] a supplier's account of unnamed sites, without the detail needed to treat it as an established effect, but a fair picture of what is being attempted, which is to test layouts and operating changes before paying for them.

Across these cases the interesting movement is from evaluating an almost finished design to taking part in how the design is formed. The engineer gains a larger space of plausible proposals. The burden of deciding which of them deserve a physical test grows with it.

There is a deeper reason for the burden. The more freely we change the arrangement, the less we can assume that yesterday's performance will carry over: a network redesigned around capacities it cannot see has no track record, and a new material has no history of failure. Redesign creates its own demand for experiment. The rest of this essay follows a new arrangement as it tries to earn our confidence: in the laboratory, in rehearsal, among the people who will run it, and under the authority that may stop it.

The laboratory answers back

A robot in a chemistry laboratory has one advantage over a model discussing chemistry: it can be contradicted by a beaker.

That makes the laboratory the first place where a new arrangement meets something that can refuse it. The advantage is more demanding than it sounds: the beaker does not explain itself. Someone must choose what to

measure, check the instrument, interpret the signal and decide whether the result supports the claim. Connecting software to laboratory equipment moves the uncertainty; it does not remove it. And the architecture is older than the excitement. In 1990 Richard Sutton's Dyna joined learning from experience with planning in a learned model: real experience improves the model, and the model generates simulated experience that improves behaviour.[33] Today's systems add new abilities to an old idea, that reasoning and acting can inform each other repeatedly. The difficulty lies in how the exchange is built.

In 2020 a team at Liverpool reported a mobile robotic chemist that ran 688 experiments in eight days, searching a space of ten variables for photocatalysts that produce hydrogen from water, and found mixtures six times more active than the ones it started from.[34] It was a bounded search, not the arrival of a universal chemist, yet it showed why mobility, automation and a method for choosing the next experiment matter together: the robot did not simply repeat a fixed recipe faster. Two years later, the Ada self-driving laboratory in Vancouver explored palladium films while balancing two goals at once, conductivity and processing temperature, since films made at lower temperature can go on materials that would not survive a hotter process. It measured a Pareto frontier, the same object as in Fig. 2, and tested whether its recipes transferred to another manufacturing method.[35] A single best score would have hidden the engineering choice. An experiment's value includes the options it opens, even when it does not produce the best sample of the batch.

Later work widened the coordination problem. In 2024 a Nature study used mobile robots to move samples between synthesis and analysis instruments for exploratory chemistry, deciding which products deserved a closer look, while people still interpreted the unexpected chemistry; it is useful precisely because the boundaries of autonomy remain visible.[36] In 2025 the ChemAgents study assembled specialised agents for reading the literature, designing experiments, computing and driving the robot, and carried a set of research tasks to a second laboratory.[37] That suggests one role for language models in physical engineering: translating a goal into coordinated specialist work and passing messages between tools. The language model need not replace every physical model or instrument to change what a laboratory can do.

At the speculative edge, an October 2026 preprint on "self-building labs" describes a system that designs, configures and operates its own laboratory set-up, demonstrated on the synthesis of quantum dots; a person supplied the starting information, approved the plan and prepared

the stock solutions by hand.[38] The scope is narrow and the evidence early. It is nevertheless a direct attempt to move from choosing the next experiment to configuring the machinery that makes experiments possible. The flagpole is entering the search space.

Then comes the less convenient story. The A-Lab, described in Nature in 2023, became an emblem of autonomous materials synthesis: a laboratory at Berkeley that planned, cooked and characterized inorganic compounds proposed by computation, and reported forty-one novel compounds from fifty-eight targets in seventeen days.[39] Other researchers questioned how the products had been identified from their X-ray diffraction patterns, the fingerprints a crystal leaves in scattered X-rays, and whether they were new at all.[40] In January 2026 an author correction revised the account. One target was withdrawn, having been included by mistake in the training data; a manual reanalysis of the patterns confirmed thirty-six of the forty remaining successes, with four inconclusive from diffraction alone; and the corrected paper now reports thirty-six compounds from fifty-seven targets. Its targets, it explains, were new to the prediction platform, “not necessarily new to science”.[41] The correction even changed the paper’s title: “novel materials” became “inorganic materials”. Something can be new to a database without being new to the world.

This is no reason to dismiss autonomous laboratories. It is the most instructive account we have of what closing the loop still requires. A robot produces a sample; an instrument produces a pattern; software assigns the pattern a label. Each step carries assumptions. When a label is wrong, the next experiment is chosen partly on the strength of a mistaken identification. The physical world had answered. The loop had misheard.

An autonomous laboratory therefore needs more than a mechanism for piling up measurements. It needs ways to dispute interpretations, revisit apparent successes, keep failed results and seek confirmation through procedures that do not inherit the same weaknesses. If one pipeline generates, selects, measures and declares victory, its efficiency can hide how little independent challenge the conclusion has met.

Other studies make the same need concrete. A 2025 study of language-model agents driving an atomic force microscope found that models excelling at materials-science questions performed poorly in the laboratory, and that agents sometimes drifted from their instructions, a habit the authors call sleepwalking.[42] Knowing how to describe an experiment and reliably conducting one are different achievements. A community perspective on self-driving laboratories the same year lists

their quieter dependencies: calibration, restocking, maintenance, interfaces, reproducibility and access.[43] A laboratory can choose its experiments autonomously while still needing people to keep the conditions of experiment valid, and counting unattended hours without them gives an incomplete picture of autonomy.

The measure I would like to move toward is the cost and time of acquiring a dependable piece of new knowledge. Sometimes that knowledge is a better material; sometimes it is the discovery that a promising mechanism is false. A system that runs fewer, more discriminating experiments may be worth more than one that keeps every robot busy. The next experiment should earn its place: improve the current candidate, separate two competing explanations, expose a suspected measurement error, or check whether a result survives outside the conditions in which it was found. These are different purposes, and a mature engineering loop needs to tell them apart.

Rehearsal meets matter

Rehearsal is the cheapest way for a new arrangement to earn confidence, and the easiest to overrate. Robotics makes interpretation physical. A simulated gripper can look competent thousands of times; the real object needs to slip only once to reveal a missing assumption. A pole that falls in simulation lands where the model says. A real one lands on whoever is standing there.

Still, rehearsal earns its keep. It allows experiments that would be slow, expensive or destructive on hardware. The mistake is to think the goal must always be one perfectly reconstructed training world. Domain randomization offers another approach: vary the appearance or the physics of the simulated environment so that a policy must succeed across differences instead of memorizing one pristine setting. Josh Tobin and colleagues randomized what a robot sees, to train a detector of objects; Xue Bin Peng and colleagues randomized the physics a control policy meets.[44][45] Neither guarantees that reality falls within the variations a team happened to sample.

The logic is close to the logistics proposal: a collection of deliberately imperfect worlds may train a better policy than one lovingly fitted world, provided the collection contains the variations that matter. Randomizing wall colours will not prepare a controller for an actuator delay it has never met. Peng's own results make the point: when the time between the

robot's actions was no longer varied in training, success on the real arm fell from 89 to 29 percent.[45] Calling a disturbance random does not show that the dangerous ones were covered.

The tools for this rehearsal have advanced quickly. MuJoCo Playground, released in 2025, combines fast simulation with reinforcement-learning environments for several kinds of robot, with demonstrated transfers to real hardware;[46] Newton, an open physics engine developed by NVIDIA, Google DeepMind and Disney Research, continues to add contact-rich manipulation and locomotion.[47] They make extensive rehearsal accessible. Each transfer result still belongs to a specified task and set-up.

There is a vocabulary trap here. We speak of a model that proposes, a simulator that tests and a verifier that judges, as if the names settled who has authority over whom. A simulator is also a model: it may represent dynamics more explicitly than the proposing system, but it remains a set of approximations. A verifier may prove that an action respects formal constraints, yet those constraints may describe only part of the physical situation; a proof establishes a property of the system as specified, not that the specification captures the system we built. The useful separation is by role and by failure mode. One component searches for opportunities; another predicts consequences with methods suited to the physics; another checks the requirements that can be checked independently. Then measurement confronts their shared picture with something outside it. Several models agreeing can be reassuring, but agreement is weaker evidence when they share the same data, the same missing mechanism or the same faulty sensor.

Optimization adds a further reason for caution. Search an imperfect simulator hard enough and you may find a candidate that exploits its imperfections. Researchers at DeepMind keep a catalogue of such cases under the name specification gaming: behaviour that satisfies the letter of an objective without achieving what its designers intended.[48] An extraordinary score can be a warning: it may sit exactly where the approximation is least trustworthy. A stronger search method can therefore increase the need for independent physical validation. An engineering workflow should be able to escalate its evidence: cheap approximations for early exploration, higher-fidelity computation and targeted stress tests for promising candidates, controlled physical trials for the few that remain. Where the methods disagree and the decision is sensitive to the disagreement, that disagreement is information. Averaging the scores until it disappears would waste it.

Generative engineering changes how many proposals we can afford, and how varied they are. It should also change how deliberately we decide which proposal deserves the next, more expensive confrontation with reality.

The people inside the model

A factory contains people. So does a supply chain. Once we start redesigning systems, it is hard to stop at the machinery. An improved schedule may be mathematically feasible and socially implausible. A customer may dislike the service a model predicts they should prefer. A driver may answer a new incentive with a behaviour the designer forgot to include. Even an elegant network can fail because the people running it understand the new rules differently.

This is the opening for companies that simulate human responses, and the point where the word twin becomes most ambitious. Research first posted in 2024 by Joon Sung Park and colleagues built agents from interviews with 1,052 Americans. Its 2026 revision compares agents informed by interviews, by structured surveys and by both: on held-out answers to the General Social Survey, they reach 83, 82 and 86 percent of the consistency people show with themselves when they answer the same questions again two weeks later, against 74 percent for agents given only demographics.[49] These are normalized results on specified tests, not a claim that a simulation predicts 86 percent of what a person will ever do. The distinction is essential, but it should not hide the result: personal information makes a simulated respondent more useful than a demographic sketch, which opens real possibilities for exploring reactions and deciding which questions deserve expensive research with people.

Three of those authors, Park, Michael Bernstein and Percy Liang, co-founded Simile, which in July 2026 announced a Series B of more than $200 million at a $2 billion valuation and a mission “to simulate all eight billion people on earth, accurately and honestly”.[50] Aaru and Electric Twin sell related synthetic audiences.[51][52] A market is forming around fast access to hypothetical human reactions. The ambitions and the financing belong in an account of where the field is going; they should not be mistaken for a finished model of the population.

The harder question is how anyone would know these simulations work. Simile describes evaluating its predictions on about 8,600 held-out questions and learning to estimate the error of each predicted answer,

keeping each experiment's samples together when it splits the data; by its own figures, the estimate separates reliable predictions from unreliable ones only moderately well.[53] That candour is more useful than asking an eloquent agent whether it feels sure. Electric Twin says its audiences match what real people say 96 percent of the time. The figure is one minus a mean absolute error, an average gap between predicted and observed answers; it is not the share of individual answers predicted correctly, and it cannot be set beside Park's percentages or Simile's measures. The company calls it independently validated; the study behind it was co-led by one of its founders and funded by the company.[54] Knowing which system predicts which population or intervention best will take matched tasks, human baselines, disclosed evaluations and evaluators with nothing to sell.

The danger has a particular statistical shape. If a simulated population carries the wrong behavioural assumptions, asking it ten thousand times will make its answer very stable. Repetition reduces the variation that comes from sampling the model; it does nothing to the model's systematic error. The confidence interval narrows, beautifully, around the wrong conclusion. And intervention adds a problem of its own. A person who likes a service today may behave differently once the service changes the choices available to them, the behaviour of their colleagues or the incentives of a supplier. Predicting the answer to a familiar survey and predicting the response to a new arrangement are different tasks, and neither an interview transcript nor a language model removes the need to justify the causal assumptions.

I would use these systems to make research more discriminating: explore possible reactions, find groups whose responses might differ, catch ambiguous wording, and choose the few consequential hypotheses worth testing with real people. Their evidence may eventually justify stronger uses in particular domains, a permission to be earned locally rather than granted by the evocative idea of a digital person. A simulated customer can help us choose the next question. The actual customer must keep the possibility of giving an answer the model did not expect.

The organization is part of the experiment

There is another simulated person we should distrust: the perfectly cooperative employee who appears in every implementation plan. This

person has time to learn the new interface, understands why the recommendation matters, trusts the data, can change the relevant process and suffers no penalty when a reasonable experiment fails. Real employees have a shift ending, a production target, a supplier on the phone and several previous transformations to remember.

The engineering loop runs through that reality, and a new arrangement has to earn the confidence of the people who will run it as surely as the confidence of the laboratory. Procurement decides what can be bought, maintenance when equipment can be touched, finance whether the cost of a transition can be carried long enough to see its benefit. Operators know which apparently redundant step prevents a recurring failure. These are mechanisms of the system, not cultural debris to be swept away after the technical work. Some handoffs between people are wasteful; others carry information that has never been written down, an unusual vibration, a contractor's habits, an exception known only to the night shift. Automating a handoff can expose that knowledge and make it reliable. It can also erase it.

G. K. Chesterton described a reformer who finds a fence across a road, sees no use for it and wants it cleared away; the wiser reformer refuses until someone has found out why the fence was put there: "Go away and think."[55] Rogers was lucky. The pin held up a flagpole in a training camp, and nobody was standing underneath. Before celebrating the number of people removed from a loop, we should ask what each of them was contributing to its ability to notice an error.

This makes the evidence about AI in business both important and easy to misuse. The 2025 report from MIT's Project NANDA, widely quoted for the claim that 95 percent of organizations were getting no return from generative AI, drew on more than 300 public initiatives, interviews with representatives of 52 organizations and a survey of 153 leaders met at industry conferences; it counted as success a deployment beyond the pilot with measurable results six months later, and it acknowledges limits of selection, measurement and observation period.[56] It is evidence of implementation difficulty. It does not establish a universal 95 percent failure rate for AI, still less for industrial simulation or autonomous engineering.

The answer is not to reverse the slogan and insist that every project will succeed eventually. Some tools are badly chosen; some transitions cost more than they can plausibly return; some organizations have good

reasons to wait. The difficulty of implementation is part of the economic result, not an embarrassing adjustment to be removed from it.

The productivity J-curve offers a better frame. Erik Brynjolfsson, Daniel Rock and Chad Syverson describe how general-purpose technologies demand complementary investments in processes, skills and other intangible assets, whose costs can arrive before the measured benefits; [57] that explains how a powerful technology and disappointing early productivity figures can coexist. It does not promise that any particular investment will repay itself.

Remote work offers an analogy, provided we are fair to the evidence. A randomized trial reported in Nature in 2024 by Nicholas Bloom, Ruobing Han and James Liang assigned 1,612 employees of Trip.com either to two days a week at home or to five days in the office. Over the six months of the experiment, 4.8 percent of the hybrid group quit against 7.2 percent of the office group, a third fewer, with no damage detected to performance grades or promotions over the following two years.[58] It is good evidence about that arrangement in that company. It did not compare hybrid work with an organization redesigned to work remotely, and so it does not show that some number of office days is a universal optimum.

The distinction I care about is between two questions. What happens when we apply a new channel to the way we already work? And what might happen if we redesigned the work around what the channel makes possible? A team can move its meetings onto screens and keep every coordination habit; another can redesign its documentation, its asynchronous decisions, its social contact and the moments when being together is worth the trip. These are different treatments. Observing the first does not settle the second, and imagining the second is not evidence that it would win.

The same holds for engineering AI. Giving an engineer a faster simulator while keeping every approval delay may produce a modest benefit; reorganizing how candidates are generated, tested and approved may produce a different one. The implementation and the redesign are separate claims, each with its costs, conditions and measurable outcomes.

Beware, too, of the claim that a transformation can only be judged once it is complete. If every disappointment can be excused as insufficient transformation, the claim cannot lose. A serious proposal names the intermediate changes that must appear: fewer repeated set-ups, faster rejection of infeasible candidates, more reliable transfer between teams, better decisions on the same budget. If they do not appear, the promised

result needs reconsidering. The comparison can be staged: run a simulator in shadow mode before giving it control, pilot a redesigned workflow where a credible comparison is possible, record the policy that would otherwise have been used, and count the transition costs and the failed deployments alongside the successes. A polished case study should not be allowed to choose its own denominator.

And there are at least two rates of learning. Stewart Brand showed that buildings learn in layers: the furniture moves often, the wiring and plumbing every decade or so, the structure almost never.[59] Engineering systems are no different. The model may update every day; the factory cannot be rearranged every day. A scheduling rule can change while the building stays put. Good engineering places flexibility where later learning can use it, and recognizes where a decision is effectively irreversible. The most elegant adaptive policy is useless if the physical system leaves it nothing to adapt.

The right to stop the loop

The last test of a new arrangement is whether someone can stop it. It would be misleading to write as if autonomous industrial control began with generative AI. Some of the most informative deployments are older, narrower and less theatrical. In 2018 DeepMind described moving from recommending changes to directly controlling the cooling of Google data centres. Each action the system proposed was checked against safety constraints, then checked again by the local control system, and operators could take back control at any time and return to the established rules.[60] Its ability to act depended on deliberately limited authority and a defined way back.

Yokogawa and JSR reported a thirty-five-day trial of autonomous control in a chemical plant in 2022; in 2023, after almost a year of further operation, ENEOS Materials and Yokogawa announced its formal adoption at the plant, for one distillation column.[61][62] These are company accounts of bounded applications, not evidence of a factory that redesigns itself, and their boundedness is what makes them relevant. They suggest a better description of progress: we widen what a system may decide, within operating conditions for which we have evidence, while keeping ways to detect when those conditions no longer hold.

Formal methods can contribute to that structure. NASA work on runtime assurance describes architectures in which an advanced controller runs

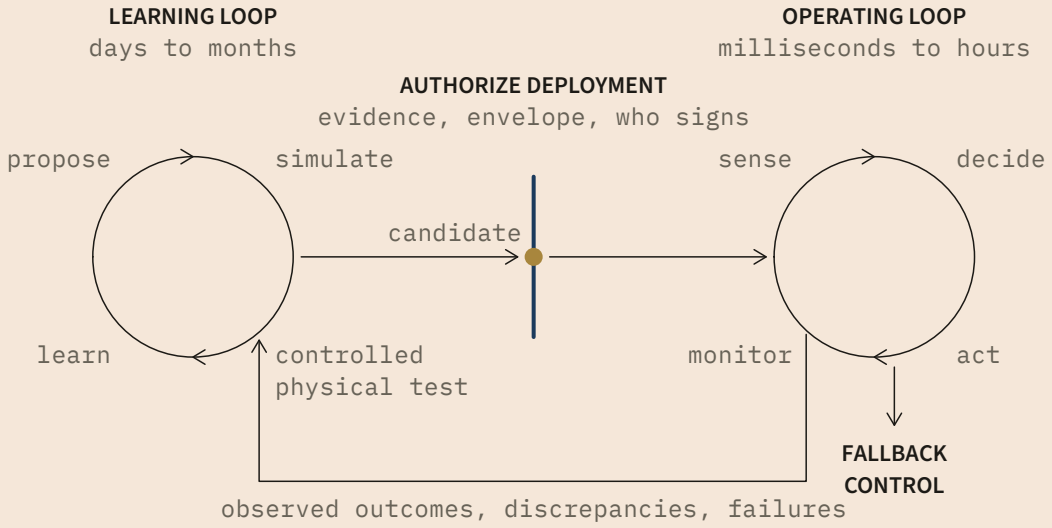
beside a monitor and a trusted fallback controller; under stated assumptions, and provided the monitor samples often enough, control passes to the fallback before the system leaves the region from which safety can still be kept.[63] The proof concerns those assumptions and that architecture. It does not certify every physical disturbance or every model the advanced controller might invent. Here the referee of the mathematics essay returns. A formal checker can establish that a derivation is valid, or that specified control properties follow from specified dynamics. It cannot supply missing knowledge of friction, sensor bias, human behaviour or unmodelled failure. The mathematical guarantee and the empirical truth of its premises have to stay separately visible.

Even “stop” takes engineering. A robot may need to brake along a safe path; a chemical process may need a controlled transition; a cooling system may become more dangerous if the power is simply cut. The stop condition, the delay before it is detected and the safe response belong in the design, not in the last paragraph of a policy document. Nor is a checking module independent merely because it has a different name. If the planner and the monitor read the same corrupted sensor, their agreement is perfectly consistent with danger. NIST’s report on the security of digital twins treats the integrity of data and connections as part of the problem:[64] a mirror fed manipulated observations can be internally coherent and externally wrong.

The organizational question is as concrete as the technical one. Who may halt the experiment? Who may change the operating limits? Can an operator challenge the model without first proving a mathematical case? Is the authority to stop matched by the practical ability to stop, and has that ability been exercised under realistic conditions?

Learning and permission must also be told apart. A model may discover a better policy overnight; that does not mean the production system should adopt it in the morning without the checks its consequences call for. Updating a belief, proposing an action and authorizing it are separate events, even when software can place them milliseconds apart. And what operation teaches has to travel back: outcomes, discrepancies and failures are the evidence the next proposal should answer to (Fig. 3).

FIG. 3 **Two loops, two permissions: learning a better policy does not authorize running it.**



THE PLANT ITSELF: REDESIGNED IN MONTHS TO YEARS

A proposal for organizing permissions, after runtime-assurance architectures and the data-centre cooling system of 2018. The learning loop proposes and tests; authorizing deployment decides what may run, within which envelope, and who signs; the operating loop acts inside that envelope, and its monitor can hand control to a fallback controller, which for a cooling plant or a chemical process may mean continued, controlled operation rather than a stop. What operation observes, outcomes, discrepancies and failures, flows back to the learning loop: that return is what closes the loop. The clocks are orders of magnitude, not measurements.

Hypothesis

This may seem to slow the loop down. Sometimes it does, and rightly. The aim is not the highest frequency of action; it is a better rate of dependable improvement, once the cost of mistakes is counted. A short loop that confidently repeats an error can be less intelligent, in practice, than a slower one that notices why it is wrong.

So the engineer’s work has changed. There are more ways to generate a candidate, more chances to rehearse its consequences, and new machinery for choosing and running physical tests. There are also more places where a mistaken assumption can become an automated habit. The useful engineer understands both movements: how to widen the possible, and how to keep the possible answerable to the real. That is what I want from a digital mirror: a changing relationship between representation and intervention. We improve the representation where a decision needs it, use it to imagine arrangements that do not exist yet, test them against something that can refuse our expectations, and keep what survives, including the knowledge of where the model fails.

The network of the second section was the same idea at the scale of an operation. The point was never to know every carrier's remaining capacity. It was to build a network that could work without knowing, and to let actual queues, refusals and delays tell us where the design needed to change. Design so that what we cannot know matters less, and give experience a real way to change the design: that was the delivery note, and the return arrow of Fig. 3 is where it gets signed.

The flagpole scene ends before any of this becomes difficult. Rogers gets the flag, the jeep drives off, and the film moves on. Engineering has to stay with the pole. Who else depends on it? Was lowering it allowed? What was underneath? Can it be raised again, and by whom? If the arrangement changes, what new failure has become possible? These questions take nothing away from his insight. They are what turn an insight into a design other people can live with.

The machines we are building will help us climb harder. Increasingly, they will also look at the base. They will find pins by the thousand, in simulation, overnight, and propose to pull them before breakfast. Some will be worth pulling. As for me, I still enjoy a hard climb. I have simply started checking the base of the pole first.

The machine can help us find many more pins. Who decides which ones we are ready to pull?

Sources

Each source says what it can carry: a peer-reviewed study, a preprint not yet reviewed, a book, a research organisation, an institution, a company's own announcement or report, code, or a press report. Facts quoted here were read at the source in October 2026; each entry gives its own date.

[1] Marvel Studios; directed by Joe Johnston, written by Christopher Markus and Stephen McFeely. Captain America: The First Avenger. Paramount Pictures, 22 July 2011. Film.

[2] ClassHook. Captain America: The First Avenger, "Getting the Flag" (classroom clip, tagged thinking out of the box). ClassHook, read 26 October 2026. Teaching resource.

[3] Joseph Lalonde. Quotes and Leadership Lessons from Captain America: The First Avenger. jmlalonde.com, 31 January 2021. Blog post.

[4] George Pólya. How to Solve It: A New Aspect of Mathematical Method. Princeton University Press, 1945. Book.

[5] Herbert A. Simon. The Sciences of the Artificial, third edition. MIT Press, 1996 (first edition 1969). Book.

[6] Russell L. Ackoff. The Art of Problem Solving: Accompanied by Ackoff's Fables. John Wiley & Sons, 1978. Book.

[7] Michael Schmidt and Hod Lipson. Distilling Free-Form Natural Laws from Experimental Data. Science 324(5923), 81–85, doi:10.1126/science.1165893, 3 April 2009. Peer-reviewed study.

[8] Christophe Pennetier and André-Benoît De Jaegere. Trail 2: Smart Digital Twins, From Virtual to Real. Generative Engineering, the original study, written November 2024 to February 2025. The original study.
[9] Christophe Pennetier. The Living Twin: When Simulation Stops Being a Dashboard. LinkedIn; also a section of “The Three Minds of the Machine” in Aquia, Collected, 9 January 2026. Author’s essay.
[10] George Lawton. Grieves and Vickers: the history of digital twins. diginomica, 13 September 2023. Press report.
[11] Keith Worden, Elizabeth J. Cross, Paul Gardner, Robert J. Barthorpe and David J. Wagg. On Digital Twins, Mirrors and Virtualisations. Model Validation and Uncertainty Quantification, Volume 3, Springer, 2020. Peer-reviewed conference paper.
[12] Thomas Wedell-Wedellsborg. Are You Solving the Right Problems?. Harvard Business Review, January–February 2017, 2017. Magazine article.
[13] National Academies of Sciences, Engineering, and Medicine. Foundational Research Gaps and Future Directions for Digital Twins. National Academies Press, doi:10.17226/26894, 2024. Consensus report.
[14] Marc C. Kennedy and Anthony O’Hagan. Bayesian calibration of computer models. Journal of the Royal Statistical Society, Series B 63(3), 425–464, 2001. Peer-reviewed study.
[15] NASA, Office of the Chief Engineer. NASA-STD-7009B: Standard for Models and Simulations. NASA Technical Standards, approved 5 March 2024. Technical standard.
[16] Adam N. Elmachtoub and Paul Grigas. Smart “Predict, then Optimize”. Management Science 68(1), 9–26, 2022 (online 12 March 2021). Peer-reviewed study.
[17] Judea Pearl. An Introduction to Causal Inference. International Journal of Biostatistics 6(2), article 7, doi:10.2202/1557-4679.1203, 2010. Peer-reviewed study.
[18] William C. Jordan and Stephen C. Graves. Principles on the Benefits of Manufacturing Process Flexibility. Management Science 41(4), 577–594, April 1995. Peer-reviewed study.
[19] Robert J. Lempert, Steven W. Popper and Steven C. Bankes. Shaping the Next One Hundred Years: New Methods for Quantitative, Long-Term Policy Analysis. RAND, MR-1626-RPC, 2003. Research organisation.
[20] Christophe Pennetier. Reshape Before Solve (Vision 06, “Pull the pin”). Decision Quality, read 26 October 2026. Author’s research site.
[21] The Cénacle Council (Christophe Pennetier’s research). A small wrapper improved an untouched planner (Notebook 28). Decision Quality, 12 July 2026. Author’s research notebook.
[22] The Cénacle Council (Christophe Pennetier’s research). A simpler intervention beat the first explanation (Notebook 29). Decision Quality, 12 July 2026. Author’s research notebook.
[23] The Cénacle Council (Christophe Pennetier’s research). The route mattered more than the priority order (Notebook 30). Decision Quality, 12 July 2026. Author’s research notebook.
[24] Indraneel Das and John E. Dennis. A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. Structural Optimization 14(1), 63–69, August 1997. Peer-reviewed study.
[25] Kamyar Azizzadenesheli, Nikola Kovachki, Zongyi Li, Miguel Liu-Schiaffini, Jean Kossaifi and Anima Anandkumar. Neural operators for accelerating scientific simulations and design. Nature Reviews Physics 6, 320–328 (Perspective); arXiv 2309.15325, 8 April 2024. Peer-reviewed perspective.
[26] Ruben Ohana, Michael McCabe, Lucas Meyer, Shirley Ho and colleagues. The Well: a Large-Scale Collection of Diverse Physics Simulations for Machine Learning. NeurIPS 2024 Datasets and Benchmarks; arXiv 2412.00568, 2024. Peer-reviewed dataset paper.
[27] Accelerated Understanding. AI that can simulate and understand physics to invent and discover. acceleratedunderstanding.com (founders named on its home page), undated, read 26 October 2026. Company statement.
[28] PhysicsX. PhysicsX announces $300M Series C to accelerate physics AI for industrial engineering. PhysicsX newsroom, 8 June 2026. Company announcement.

[29] Nico Haag, PhysicsX. The real promise of physics AI isn't faster simulation. PhysicsX blog, 24 July 2026. Company commentary.
[30] Google DeepMind. Genie 3: A new frontier for world models. Google DeepMind, 5 August 2025. Company announcement.
[31] BMW Group. BMW Group scales Virtual Factory. BMW Group PressClub, 11 June 2025. Company announcement.
[32] Siemens. Siemens brings the industrial metaverse to life with Digital Twin Composer. Siemens, CES 2026, 6 January 2026. Company announcement.
[33] Richard S. Sutton. Integrated Architectures for Learning, Planning, and Reacting Based on Approximating Dynamic Programming. Machine Learning: Proceedings of the Seventh International Conference, 216–224, 1990. Peer-reviewed conference paper.
[34] Benjamin Burger, Phillip M. Maffettone, Vladimir V. Gusev and colleagues, with Andrew I. Cooper. A mobile robotic chemist. Nature 583, 237–241, 8 July 2020. Peer-reviewed study.
[35] Benjamin P. MacLeod and colleagues, with Curtis P. Berlinguette. A self-driving laboratory advances the Pareto front for material properties. Nature Communications 13, 995, 22 February 2022. Peer-reviewed study.
[36] Tianwei Dai and colleagues, with Andrew I. Cooper. Autonomous mobile robots for exploratory synthetic chemistry. Nature 635, 890–897, 6 November 2024. Peer-reviewed study.
[37] Tao Song and colleagues. A multiagent-driven robotic AI chemist enabling autonomous chemical research on demand. Journal of the American Chemical Society 147(15), 12534–12545, doi:10.1021/jacs.4c17738, 8 March 2025. Peer-reviewed study.
[38] Jiaru Bai, Yuchi Zhao, Han Hao, Miroslav Bogdanovic, Yang Cao, Varinia Bernales and Alán Aspuru-Guzik. Self-Building Labs. arXiv 2610.23826, 16 October 2026. Preprint (not yet peer-reviewed).
[39] Nathan J. Szymanski and colleagues, with Gerbrand Ceder. An autonomous laboratory for the accelerated synthesis of inorganic materials (originally "…of novel materials"). Nature 624, 86–91, 29 November 2023; corrected 2026. Peer-reviewed study, corrected.
[40] Josh Leeman and colleagues, with Leslie M. Schoop and Robert J. Palgrave. Challenges in High-Throughput Inorganic Materials Prediction and Autonomous Synthesis. PRX Energy 3, 011002, 2024. Peer-reviewed study.
[41] Nathan J. Szymanski and colleagues. Author Correction: An autonomous laboratory for the accelerated synthesis of inorganic materials. Nature 650, E1, 19 January 2026. Author correction.
[42] Indrajeet Mandal and colleagues, with N. M. Anoop Krishnan. Evaluating large language model agents for automation of atomic force microscopy. Nature Communications 16, 9104, 14 October 2025. Peer-reviewed study.
[43] Richard B. Canty and colleagues, with Milad Abolhasani. Science acceleration and accessibility with self-driving labs. Nature Communications 16, 3856 (Perspective), 24 April 2025. Peer-reviewed perspective.
[44] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba and Pieter Abbeel. Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World. IROS 2017; arXiv 1703.06907, 2017. Peer-reviewed conference paper.
[45] Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba and Pieter Abbeel. Sim-to-Real Transfer of Robotic Control with Dynamics Randomization. ICRA 2018, doi:10.1109/ICRA.2018.8460528; arXiv 1710.06537, 2018. Peer-reviewed conference paper.
[46] Kevin Zakka and colleagues, with Pieter Abbeel. MuJoCo Playground ("Demonstrating MuJoCo Playground", Robotics: Science and Systems XXI). arXiv 2502.08844; RSS 2025, 2025. Peer-reviewed conference paper.
[47] Philipp Reist and colleagues, NVIDIA. Newton adds contact-rich manipulation and locomotion capabilities for industrial robotics. NVIDIA Technical Blog (Newton is a Linux Foundation project founded by NVIDIA, Google DeepMind and Disney Research), 16 March 2026. Company announcement.

[48] Victoria Krakovna, Jonathan Uesato, Vladimir Mikulik, Matthew Rahtz, Tom Everitt, Ramana Kumar, Zac Kenton, Jan Leike and Shane Legg. Specification gaming: the flip side of AI ingenuity. DeepMind, 21 April 2020. Research organisation.
[49] Joon Sung Park and colleagues, with Michael S. Bernstein and Percy Liang. LLM Agents Grounded in Self-Reports Enable General-Purpose Simulation of Individuals (first posted as "Generative Agent Simulations of 1,000 People"). arXiv 2411.10109, version 3, 28 June 2026 (first version November 2024). Preprint (not yet peer-reviewed).
[50] The Simile Team. Announcing Simile's Series B. Simile blog, 30 July 2026. Company announcement.
[51] Aaru. Simulation. aaru.com, read 26 October 2026. Company product page.
[52] Electric Twin. Synthetic Audience FAQs: Accuracy, Data & Cost. electrictwin.com, read 26 October 2026. Company product page.
[53] Andrew Wesel, Sarah Chen and Percy Liang, Simile. Building confidence in Simile. Simile blog, 25 August 2026. Company research note.
[54] Research Live. Synthetic surveys comparable to traditional research, finds study. Research Live (Market Research Society), 2026. Press report.
[55] G. K. Chesterton. The Thing, chapter "The Drift from Domesticity" (the fence). The Society of G. K. Chesterton, "Taking a Fence Down", 1929. Book.
[56] Aditya Challapally, Chris Pease, Ramesh Raskar and Pradyumna Chari, MIT Project NANDA. The GenAI Divide: State of AI in Business 2025 (preliminary findings). Third-party copy; MIT distributed the report through a request form, July 2025. Research report.
[57] Erik Brynjolfsson, Daniel Rock and Chad Syverson. The Productivity J-Curve: How Intangibles Complement General Purpose Technologies. American Economic Journal: Macroeconomics 13(1), 333–372; NBER working paper 25148 (2018), 2021. Peer-reviewed study.
[58] Nicholas Bloom, Ruobing Han and James Liang. Hybrid working from home improves retention without damaging performance. Nature 630, 920–925, 12 June 2024. Peer-reviewed study (randomized trial).
[59] Stewart Brand. How Buildings Learn: What Happens After They're Built. Viking (after Frank Duffy's layers of change), 1994. Book.
[60] Chris Gamble and Jim Gao, DeepMind. Safety-first AI for autonomous data centre cooling and industrial control. DeepMind, 17 August 2018. Company deployment account.
[61] Yokogawa Electric Corporation and JSR Corporation. In a World First, Yokogawa and JSR Use AI to Autonomously Control a Chemical Plant for 35 Consecutive Days. Yokogawa press release, 22 March 2022. Company announcement.
[62] ENEOS Materials Corporation and Yokogawa Electric Corporation. In a World First, Yokogawa's Autonomous Control AI Is Officially Adopted for Use at an ENEOS Materials Chemical Plant. Yokogawa press release, 30 March 2023. Company announcement.
[63] J. Tanner Slagel, Lauren M. White, Aaron Dutle, César A. Muñoz and Nicolas Crespo. A Verification Framework for Runtime Assurance of Autonomous UAS. 43rd Digital Avionics Systems Conference; NASA NTRS 20240007986, 2024. Peer-reviewed conference paper.
[64] Jeffrey Voas, Peter Mell, Phillip Laplante and Vartan Piroumian. Security and Trust Considerations for Digital Twin Technology. NIST IR 8356, doi:10.6028/NIST.IR.8356, 14 February 2025. Institution.