histories

MDPI

Article

The Functional-Delegability Frontier: Rethinking the Industrial Revolution as a Long-Run Historical Process

Armando Sternieri

Energee3 S.r.L., 42121 Reggio Emilia, Italy; armando.sternieri@energee3.com

Abstract

The Industrial Revolution is commonly periodized through mechanization, steam, electricity, mass production, computing, digital networks, and artificial intelligence. This article proposes an alternative variable of historical observation: the functional-delegability frontier, defined as the historically demonstrated set of functions technically executable by technical systems under identifiable operating conditions. A function is specified independently of its mechanism as F = (I, O, C): admissible inputs, required output, and operating conditions. Analysis distinguishes extensive and intensive expansion, functional flexibility, recursive expansion, and recursive depth. A comparison of employment-based, energy-based, and techno-economic interpretations clarifies their different criteria of historical change. Applications to machine tools, automatic programming, circuit simulation, and code generation identify what the evidence supports and what would require reclassification. The cases document cross-sectoral reuse of production capabilities and the participation of computing in developing subsequent technical systems; they support, without conclusively establishing, the hypothesis of one still-unfolding Industrial Revolution. This interpretation concerns functional reallocation and cumulative dependence across technological discontinuities, not the absence of internal phases. Contemporary evidence is separated from prospective claims, and environmental constraints are treated as conditions of sustainable implementation rather than proof of technical impossibility. The framework identifies technical possibility, not adoption, profitability, labor displacement, or economic destiny.

Keywords: industrial revolution; history of technology; technological change; automation; functional delegability; computing; artificial intelligence; industrialization

1. Introduction

Histories of industrialization are usually organized around technologies, sectors, energy regimes, production systems, or successive 'industrial revolutions' (Landes 1969; Mokyr 1990; Hughes 1983; Ceruzzi 2003). Mechanization and steam, electricity and mass production, and more recently computing, digital networks, and artificial intelligence provide powerful markers of discontinuity. These periodizations identify real changes. Yet they can also obscure a different continuity: the changing boundary between functions performed by human beings and functions that technical systems can execute.

This paper proposes that boundary as an object of historical analysis. It calls it the functional-delegability frontier: the historically demonstrated set of functions whose execution by technical systems is technically feasible under identifiable conditions. Technical delegability is analytically distinct from subsequent adoption: a demonstrated function may remain predominantly human-performed for long periods.

Check for updates

Academic Editor: Matteo Valleriani

Received: 26 August 2026
Revised: 15 October 2026
Accepted: 2 November 2026
Published: 4 November 2026

Copyright: © 2026 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

The proposed variable differs from a history of automation narrowly understood. "Automation" often bundles together very different transformations: a machine may execute a genuinely new function, perform a familiar function across a broader range of conditions, or provide a general-purpose architecture through which many already feasible functions can be realized. These changes need not have the same historical meaning. Watt's governor, electrification, Hollerith tabulation, general-purpose computing, and machine learning are all historically consequential, but they need not represent the same kind of change in functional delegability.

The paper advances a further historical hypothesis. Functional delegation itself long predates industrialization. What distinguishes the Industrial Revolution may therefore lie not in the first appearance of machine-executable functions, but in a transformation of the dynamics through which technical capabilities accumulate and interact. The paper asks whether mechanization, automatic control, information-processing machinery, computing, digital technologies, and artificial intelligence can be interpreted as heterogeneous technological transformations within a single long-run industrial process. Separately, it examines whether recursive depth has increased as previously delegated functions perform substantive parts of the development of further technical capabilities.

This claim is not definitional. Industrialization is treated as the observable historical process of sustained technological, productive, organizational, and economic transformation. The "single Industrial Revolution" is an interpretive hypothesis concerning the historical unity of that process. The framework can therefore remain useful even if that stronger periodization hypothesis is rejected. It does not exclude phases defined by shifts in employment, energy use, or technological organization; it asks whether the capabilities involved in those phases remain part of a historically connected process.

Section 2 locates the proposal relative to alternative historical approaches, information and control, general-purpose technologies, and task-based economics. Section 3 defines the functional unit and distinguishes extensive expansion, intensive expansion, functional flexibility, and recursive expansion. Section 4 applies these distinctions to historical and boundary cases. Section 5 develops the single-Industrial-Revolution hypothesis through the documented connections among capabilities. Section 6 examines computing, digital systems, and contemporary AI, including their material conditions. Section 7 discusses limits, empirical checks, and a research agenda. Section 8 concludes.

2. Historiographical Context and Analytical Gap

2.1. Industrialization, Technological Discontinuity, and Useful Knowledge

Economic and technological histories have long debated what made modern industrialization distinctive and why it became sustained. Explanations emphasize combinations of energy, relative prices, institutions, markets, human capital, scientific culture, and technical knowledge (Landes 1969; Allen 2009). Mokyr's account is especially relevant because it treats modern growth as dependent on changes in 'useful knowledge' and on the interaction between propositional and prescriptive knowledge (Mokyr 2002, 2005). This article does not offer an alternative explanation for why industrialization emerged. It asks what happens to the space of technically executable functions once technological change becomes sustained.

The distinction matters for periodization. Perez and Murray Leach (2022, p. 3) explicitly relate divergent interpretations to the questions, premises, and interactions selected for analysis. More specifically, Clark (1940, p. 7; chap. V) follows changes in the sectoral shares of employment, Wrigley (2010, chap. 9) explains the English transition beyond the energy constraints of an organic economy, and Perez (2002, 2009) identifies successive technological revolutions and techno-economic paradigms. Table 1 compares these approaches with

the present proposal. They identify different historical changes; none is treated here as an incorrect measurement of functional delegability.

Table 1. Historical approaches, criteria of change, and the functional-delegability perspective.

Approach and Source	Question and Observed Property	Historical Articulation	Coverage and Relation to This Article
Sectoral employment Clark (1940, p. 7; chap. V)	How does the distribution of labor change? Shares of employment in primary activities (mainly agriculture), secondary activities (including manufacturing and construction), and tertiary activities (services).	Relative contraction of primary employment; secondary employment rising to a peak and then declining relative to tertiary activity, with different turning points across countries.	Identifies structural and occupational change, not the first technical execution of functions. Clark's secondary category includes manufacturing, mining, and construction; these shifts are not labelled separate Industrial Revolutions here.
Energy and sustained growth Wrigley (2010, chap. 9)	Which energy constraints limit productive expansion? An "organic economy" relies largely on renewable flows tied to land and living materials; fossil fuels such as coal allow additional energy inputs.	The English transition beyond the constraints of an organic economy through the use of coal.	Explains an enabling condition of sustained growth. This is not a universal sequence of fuels, nor a criterion for the technical executability of each function.
Technological revolutions and techno-economic paradigms Perez (2002, 2009, Table 1)	Which linked technologies, infrastructures, and organizational practices transform the economy? A "techno-economic paradigm" names that wider configuration, not just one invention.	Five revolutions: early industrial mechanization; steam and railways; steel, electricity, and heavy engineering; oil, automobiles, and mass production; information and telecommunications.	Identifies systemic technological and economic reconfigurations. Their existence does not by itself settle whether functional reallocation across them belongs to one longer process.
Functional delegability Present article, Sections 3–5	Which functions are technically executable under specified inputs, outputs, and operating conditions, and how do successive capabilities depend on earlier ones?	A proposed long-run process containing extensive and intensive changes, increased flexibility, and episodes of recursive participation.	A historical hypothesis about functional and cumulative connections, not an alternative estimate of employment or energy use. Unity allows internal phases, uneven rates, and discontinuities.

The first three rows summarize the cited authors within their respective historical scopes; the final row states this article's hypothesis. The comparison does not establish synchronized stages across countries. It separates the criteria used to identify phases before examining the historical grounds for the functional interpretation.

2.2. Information Processing and Control

Beniger's *The Control Revolution* provides an important bridge between industrial and information history. Beniger argues that the acceleration and increasing complexity of material processes generated crises of control that stimulated new arrangements for collecting, processing, communicating, and acting upon information (Beniger 1986). This makes information processing endogenous to industrial expansion rather than an entirely separate post-industrial phenomenon.

The present framework shares that long-run perspective but uses a different organizing variable. Control is treated as one family of functions among others. The question is not only

how societies solve control problems, but how the set of functions that can be technically executed changes across physical, regulatory, informational, classificatory, predictive, and other domains.

Histories of electrification, managerial communication, and cybernetics further show that technical functions acquire historical significance within infrastructures, organizations, and systems of knowledge (Chandler 1977; Hughes 1983; Yates 1989; Kline 2015). Functional-delegability analysis does not replace these accounts. It provisionally brackets social embedding in order to identify a technical boundary, while treating diffusion and systemic consequences as distinct historical questions.

2.3. General-Purpose Technologies and Recombinant Innovation

The literature on general-purpose technologies emphasizes pervasiveness, continuing improvement, and innovation complementarities (Bresnahan and Trajtenberg 1995). Steam, electricity, and information technology are often analyzed as technologies whose effects propagate across sectors. Related accounts of cumulative and recombinant innovation emphasize how new technologies are assembled from existing capabilities and how technological components become building blocks for further invention (Rosenberg 1963, 1982; Arthur 2009).

There is clear overlap with the concepts developed below, but the unit of analysis differs. GPT theory classifies technologies. Functional-delegability analysis classifies changes in technically executable functions. A pervasive technology need not create a new technically executable function, while a narrow technology may move the functional frontier without becoming a GPT. Recursive expansion is narrower than generic cumulativeness: it requires technically delegated functions themselves to perform substantive parts of technological development.

2.4. Tasks, Automation, and the Human–Machine Boundary

Task-based economics studies how technologies change the allocation of tasks between labor and capital and the consequences for productivity, wages, and employment (Autor et al. 2003; Acemoglu and Restrepo 2019a, 2019b). This literature is indispensable for understanding how technical capabilities, economic incentives, and the creation of tasks interact.

Functional delegability addresses a logically prior question: has technical execution of a specified function been demonstrated at all? A movement of the technical frontier can occur before the function becomes an economically relevant production task. A laboratory or experimental demonstration may therefore matter historically even when its immediate economic significance is negligible. The distinction separates the emergence of technical possibility from its subsequent incorporation into production and organizational practice. Its historical relevance is not that every demonstration transforms society, but that it identifies a change in available capabilities whose subsequent use, recombination, or non-adoption can be investigated separately.

2.5. Contribution

The contribution is twofold. Analytically, the paper proposes functional delegability as a historical variable distinct from technological performance, diffusion, economic task allocation, and the importance of particular artifacts. Historically, it examines whether following this variable reveals cumulative connections across technological discontinuities conventionally used to divide industrialization into successive revolutions. The second contribution is an interpretive hypothesis assessed through historical contrasts, not a consequence of the definitions or a finding that invalidates the alternative approaches in Table 1.

What the framework adds to the approaches compared is a mechanism-independent distinction among newly executable functions, enlarged execution domains, architectural flexibility, and delegated participation in the production of further technical capabilities. These classifications can be applied across the historical phases identified by other criteria without supplanting the economic or institutional questions those approaches address.

2.6. Method: Comparative Historical Analysis and Case Selection

The article uses comparative historical analysis as a theory-building strategy rather than attempting an exhaustive history of industrialization (Mahoney and Rueschemeyer 2003). Cases are selected for analytical variation across periods and functional domains. Pre-industrial automatic regulation tests whether delegation is uniquely industrial; textile machinery and machine tools distinguish intensive change from recursive participation; Jacquard mechanisms, punched-card tabulation, and automatic switching test the difference between technical demonstration and commercial diffusion; general-purpose computing tests functional flexibility; and robotics and machine learning test mixed classifications.

Each case is examined through the same questions. What input domain, output condition, and operating conditions define the function? Was comparable technical execution demonstrated earlier? Did the later system broaden an existing function, combine functions through a common architecture, or allow delegated functions to participate materially in producing further technical capabilities? Evidence is drawn primarily from established histories of technology and economic history, supplemented by technical histories and institutional records where they document concrete milestones. Because entry into the frontier is a historical priority claim, extensive classifications are treated as provisional and carry a higher evidentiary burden than intensive ones.

The cases are therefore neither a representative sample nor a basis for estimating the frequency or economic weight of delegation. They are contrast and boundary cases chosen to expose classification errors and possible falsifiers. The long-run interpretation is evaluated separately against two conditions: continuing reallocation of executable functions between human and technical components, and cumulative dependence of later systems on capabilities generated in earlier industrial development.

The evidence serves two distinct purposes. Functional reconstructions compare specified operations, their antecedents, and the role of human intervention. Historical interpretation then examines whether capabilities were actually reused in subsequent development and what their removal would have changed. Section 5 relates these functional reconstructions to the long-run hypothesis. Institutional histories are distinguished from contemporary technical reports and experimental studies. The procedures in Section 7.7 specify further checks, not independent replications or a representative test of the entire industrial process.

During manuscript revision, ChatGPT (OpenAI, GPT-5.6 Sol) was used as an assistive tool for source retrieval and comparison, drafting and restructuring passages, language editing, and consistency checks.

3. The Functional-Delegability Framework

3.1. Definitions

The framework distinguishes different dimensions of functional delegability and technological change. Table 2 summarizes the main concepts, their definitions, and the analytical questions used to classify historical cases. These distinctions provide the basis for the functional analysis developed in the following sections.

Table 2. Definitions and analytical questions.

Concept	Definition	Analytical Question
Function	Transformation from an admissible input domain to a required output condition under specified operating conditions: F = (I, O, C).	What exactly is being executed?
Functional delegability	Demonstrated technical feasibility of executing F without a human performing F itself during execution.	Can a technical system perform the function?
Functional-delegability frontier	Boundary of the historically demonstrated set of technically executable functions.	What is technically delegable at time t?
Extensive expansion	Entry of a previously non-demonstrated function into technical executability.	Has a new function entered the frontier?
Intensive expansion	Enlargement of the domain or conditions over which an already delegable function can be executed.	How far has an existing technical capability expanded?
Functional flexibility	Range and diversity of functions executable through a common architecture without substantial physical reconfiguration.	How functionally reconfigurable is the architecture?
Recursive expansion	Delegated functions materially participate in developing additional or improved technical capabilities.	Does existing delegation contribute to further technical development?
Recursive depth	Degree and functional complexity of delegated participation in technological development.	How deeply has technical execution entered the innovation process?

3.2. Defining a Function: F = (I, O, C)

A central methodological problem is functional resolution. Descriptions such as "communication," "calculation," or "recognition" are too broad to support meaningful historical comparison. Conversely, functions defined through the specific operations of a particular machine make novelty tautological. The framework therefore defines a function independently of the mechanism performing it.

A function is a transformation from a specified admissible input domain (I) to a required output condition (O) under specified operating conditions (C). The mechanism by which the transformation is achieved is deliberately excluded. This allows historically different human and technical configurations to be compared without assuming technological similarity.

Operating conditions include only those conditions necessary to determine functional comparability: required human intervention, relevant environmental conditions, admissible input variation, error tolerance, and reliability requirements. Cost, profitability, and diffusion are excluded because they concern economic feasibility and adoption rather than technical delegability.

For example, the relevant function in automatic telephone switching is not "operating a Strowger switch." It is the transformation from addressing information specifying a destination to establishment of the corresponding communication path. A human switchboard operator and an automatic switching system can therefore be compared functionally despite radically different mechanisms.

The unit of technical execution need not be a single artifact. It may be a configured system of machines, software, infrastructure, and specified human interventions. Its boundary is functional: it includes the components required to produce O from I under C. If human intervention performs the substantive transformation rather than establishing or supervising the stated operating conditions, the function has not been fully delegated at that level of description.

For probabilistic systems, the input population, admissible error, sampling allowance, and permitted human correction must be stated before judging success. A favorable output alone does not establish reliable execution over the declared domain. These requirements specify C; they do not introduce a different definition of functional delegability.

3.3. The Frontier and Its Movement

At historical time t, the functional-delegability frontier is the boundary of the set of functions for which technical execution has been demonstrated. 'Frontier' does not imply a one-dimensional ordering. Functional space is multidimensional and movement can occur independently across domains. Nor is the frontier an autonomous property of artifacts: demonstrations are historically produced within laboratories, firms, public institutions, infrastructures, standards, and organized bodies of technical practice. The framework isolates technical executability as its variable without treating that executability as socially or institutionally self-generated.

Extensive expansion occurs when a function not previously demonstrated as technically executable enters the frontier. Such claims carry a high evidentiary burden because they require investigation of plausible predecessors. They should therefore be treated as provisional historical priority claims rather than absolute statements.

Intensive expansion occurs when a function already within the frontier becomes executable over a broader input domain, under a broader range of operating conditions, at greater scale or complexity, with higher precision, or with reduced dependence on human intervention, while retaining the identity of the input–output transformation.

3.4. Functional Flexibility

Functional flexibility is the range and diversity of distinct functions executable through a common technical architecture without substantial physical reconfiguration. An increase in flexibility does not necessarily move the external frontier. If A, B, and C are already technically executable through separate machines, a general-purpose system capable of performing all three may leave the feasible set unchanged while radically reducing the physical cost of reconfiguration. General-purpose computing is the strongest historical example examined here. Flexibility can facilitate later frontier movement, but it is not itself defined as frontier expansion.

3.5. Recursive Expansion and Recursive Depth

Recursive expansion concerns the relationship between delegated functions and subsequent technological development. It occurs when technically delegated functions perform substantive parts of the processes through which additional or improved technical capabilities are developed.

Recursive participation requires more than the presence of a technical artifact or a generic enabling service. A delegated function counts as recursively relevant when it performs an operation constitutive of designing, generating, testing, evaluating, or producing an additional or improved technical capability, and when removing it would materially alter the feasible set, speed, complexity, or resource requirements of that development process. Both conditions are required.

Electronic design automation provides a positive example: delegated simulation, verification, placement, and routing perform substantive operations in producing new integrated circuits and enable designs that would otherwise be infeasible or radically more costly. Automatic climate control in an engineering office provides a negative contrast. It may support the people developing a technology, but unless control of the environment is itself constitutive of the relevant experiment or production process, it does not execute a

substantive operation in the production of the new technical capability. This distinction separates recursive participation from general infrastructural support.

Recursive depth refers to the degree and functional complexity of such participation. The distinction can be illustrated by a sequence from tools that amplify human action, to delegated calculation, delegated design operations, delegated simulation of candidate systems, and potentially delegated generation and evaluation of technical alternatives. The sequence is not assumed to be universal or monotonic. It identifies a dimension that can be investigated historically.

Recursive depth should not be equated with intelligence, autonomy, or general technological sophistication. It is a relational property of a delegated function's position within the production of subsequent technical capability. A specialized design-automation system may therefore have greater recursive relevance than a more apparently intelligent system that does not materially participate in technological development.

4. Comparative Historical Evidence and Boundary Cases

4.1. Pre-Industrial Antecedents

Any interpretation based on functional delegation must begin by rejecting a simple pre-industrial/industrial binary. Technical systems capable of performing functions without continuous direct human execution existed long before the eighteenth century. Ctesibius of Alexandria is associated with an accurate water clock and automatic mechanisms; histories of automatic control identify ancient water-level and flow regulation as early antecedents of feedback control (Toomer 2015; Bissell 2009). Medieval Arabic mechanical engineering likewise includes a substantial tradition of water clocks, automata, pumps, fountains, and related "fine technology" (Hill 1991). Chinese engineering traditions developed sophisticated combinations of hydraulic, astronomical, and mechanical devices (Needham 1965).

Song China provides an especially important test for any interpretation that equates pre-industrial technology with isolated invention. Su Song's eleventh-century astronomical clock tower integrated water power, regulated motion, gearing, astronomical representation, timekeeping, and sequences of mechanical action. Chinese technological traditions also accumulated across metallurgy, printing, hydraulic engineering, navigation, and other domains (Needham 1965).

The implication is methodological. Occurrence, accumulation, cross-domain interaction, and even recursive technical relationships are not unique to industrialization. The historical hypothesis must therefore concern differences in degree, scope, persistence, and especially the functional depth and systemic importance of recursive participation—not the first existence of sophisticated machinery.

4.2. Mechanical Execution, Textile Machinery, and Machine Tools

Textile mechanization provides familiar examples of functions shifting from direct human execution to technical systems, but the frontier classification depends on the function specified. Spinning and weaving has long involved tools and machines; eighteenth-century inventions often expanded scale, speed, coordination, and the range of conditions under which already mechanized transformations could be performed. Historical accounts of mechanization likewise show long sequences of adaptation rather than isolated invention (Landes 1969). Many such changes are therefore better treated as intensive expansion than as unqualified first delegations.

Machine tools are more important for the paper's dynamic argument. Precision lathes, boring machines, planers, milling machines, and related systems increased the capability to produce other machines with repeatable geometries and tighter tolerances. Rosenberg (1963, pp. 428–31) traces both the transfer of machining capabilities from arms manufacture to

other industries and the development of the automatic turret lathe. The latter reduced direct human execution within a prepared machining cycle, rather than making machining as a whole possible for the first time. In the present framework, such an operation participates recursively when it contributes to developing machinery with new or improved technical capabilities; routine replication of an unchanged product is not sufficient evidence of recursive expansion.

4.3. Regulation and Control

Automatic regulation illustrates why technological importance and extensive expansion must be separated. Watt's centrifugal governor became an emblematic industrial control device, but automatic regulation had much earlier technical antecedents. The governor is therefore more defensibly interpreted as intensive expansion: regulation became applicable to increasingly important industrial processes, at greater scale and under operating conditions central to mechanized production. The case also supports a broader point made by Beniger (1986): the acceleration of material processes generated increasing demands for information and control inside industrialization rather than after it.

4.4. Encoded Control: Jacquard

The Jacquard mechanism is used here as evidence of encoded control, not as a simplistic "precursor of the computer." Punched or otherwise encoded media were used to determine sequences of warp selection, separating aspects of pattern specification from continuous manual selection. Historical accounts of sequence control trace a genealogy through Bouchon, Falcon, and Vaucanson before Jacquard's commercially successful synthesis and refinement (Randell 1994; Garcia 2018). Jacquard is therefore not treated as an uncontested extensive event. Its significance lies in the increasingly flexible and reusable relationship between encoded specification and machine action.

4.5. Machine-Readable Administrative Information: Hollerith

Hollerith's punched-card system for the 1890 U.S. Census encoded individual characteristics on cards and supported counting and cross-tabulation across attributes (Truesdell 1965). Its historical significance is clear, but the frontier classification is deliberately cautious. Counting, sorting, and calculation had technical antecedents. Hollerith therefore provides strong evidence of intensive expansion in machine-readable administrative information processing: encoded records could be processed at a scale and combinatorial range that substantially altered administrative data work. Whether particular constituent operations represent extensive expansion requires finer comparison with earlier calculating and tabulating technologies and is not assumed here.

4.6. Automatic Telephone Switching

Automatic telephone switching illustrates the evidentiary difficulty of extensive classification. Early telephone networks depended on operators to establish connections between subscribers. A caller supplied destination information; the operator identified and physically configured the required path. The function can be specified independently of the switching mechanism as destination information → establishment of the corresponding communication path.

Strowger's step-by-step system, patented in 1891 and commercially installed at La Porte, Indiana, in 1892, transferred connection selection from human operators to technical execution at operational scale. It did not, however, emerge without technical antecedents. Oscar Myers's Bell System history identifies the Connolly and McTighe "girlless" telephone system, patented in 1879, as a forerunner of step-by-step switching (Myers 1952).

The case therefore should not be used to identify Strowger itself as an uncontested extensive frontier event. Instead, it demonstrates why commercial breakthrough and historical frontier entry must be distinguished. Earlier experimental systems can establish technical feasibility before a technology becomes reliable, scalable, or commercially important. Automatic switching is thus a methodological case in which reconstruction of predecessors changes the frontier classification.

4.7. General-Purpose Computing

Electronic computing should not be treated as the first technical execution of calculation or information processing. Its distinctive importance lies partly in the extraordinary increase in functional flexibility produced by programmability. A common architecture can implement heterogeneous transformations through changes in software rather than wholesale physical reconstruction. Histories of computing document this transition from specialized calculating systems to increasingly general and networked architectures (Ceruzzi 2003).

Computing provides documented instances of recursive participation before contemporary AI. The IEEE milestone identifies Grace Hopper's A-0 system of 1951–1952 as a linker/loader and a utility for managing subroutines (IEEE History Center 2024). This is a bounded operation in software construction, not the delegation of programming as a whole. Numerical control and the Automatically Programmed Tools (APT) programming language brought encoded and computer-generated instructions into machine–tool operation (Ross 1978). Electronic design automation subsequently incorporated simulation, verification, synthesis, test generation, placement, and routing (Wang et al. 2009). Historical documentation of early integrated-circuit design also records computer-based design checking and conversion of design information into mask patterns during the 1960s (Computer History Museum n.d.). These records identify operations through which computing re-entered the production of further technical systems.

A contemporary technical report makes the simulation case more concrete. Nagel and Pederson (1973, §§I–III) describe SPICE as an improvement on CANCER, an earlier circuit-simulation program, with revised device models and an additional flicker-noise representation. Its input is a circuit description with component models and analysis conditions; its output is a calculated electrical response. The authors identify the impracticality of manual analysis for integrated circuits and report use in instruction and research. Simulation therefore executes a design-evaluation operation while leaving physical validation and the interpretation of results outside that function. The documented extensions support an intensive comparison with CANCER, not a claim that SPICE originated circuit simulation.

4.8. Industrial Robotics and Learning-Based Systems

Industrial robotics combines intensive expansion and functional flexibility. The relevant frontier movement is application-specific: some robotized operations reproduce transformations already technically executable by specialized machinery, while others extend technical execution across broader spatial, sensory, or manipulation conditions. Reprogrammability increases the range of tasks realizable through a common architecture. The history of numerical control and industrial automation also cautions against treating technical capability as independent of labor, managerial, and institutional choices (Noble 1984).

Machine learning similarly resists a single classification. Learning-based systems can broaden the input domains over which classification, prediction, recognition, and transformation functions are technically feasible, potentially producing intensive expansion. Extensive claims require historical comparison of the specified function. The code-generation case in Section 6.4 separates a dated demonstration of capability from claims about recursive participation, economic effects, or future development.

4.9. Negative and Boundary Cases

These examples concern familiar infrastructures: electricity powers equipment and permits factories to be reorganized; telegraphy sends messages over long distances; shipping containers standardize the handling of freight across transport modes; and the Internet connects computers and services. Each can transform economic activity without necessarily demonstrating a previously non-executable function. Table 3 summarizes these boundary cases and their implications for identifying frontier movement.

Table 3. Boundary cases: historical importance without an automatic inference of frontier movement.

Technology	Major Historical Transformation	Why Frontier Movement Does Not Automatically Follow
Electricity	Energy distribution, factory organization, flexibility	Changes energy source and organization without necessarily creating a newly executable function.
Telegraphy	Speed and geographical reach of communication	Amplifies transmission; interpretation and action may remain human.
Containerization	Logistics, standardization, intermodal transport	Reorganizes interfaces and coordination rather than necessarily adding technical execution of a new function.
Internet	Connectivity, distribution, coordination	Enables combinations and distribution of delegated functions, but connectivity itself is not equivalent to delegation.

These boundary cases are important because the framework must be capable of saying "no." A technology can transform productivity, geography, organization, or market structure without necessarily moving the functional-delegability frontier. Conversely, a relatively narrow technical development can move the frontier without becoming a dominant general-purpose technology. The contrast cases thus prevent economic importance from being used as a proxy for functional novelty. They delimit the framework rather than independently confirming the single-process hypothesis. Table 4 complements these negative contrasts by showing how evidence supports, qualifies, or could overturn a positive classification.

The four cases concern different operations. An automatic turret lathe machines a workpiece by carrying out a prepared sequence of cutting operations with less direct manual control. A-0 was an early programming system that assembled already written subroutines into a program, rather than writing the algorithms itself. SPICE, in its original Berkeley version (SPICE1), is a computer program that calculates a circuit's expected electrical behavior from a description of its components, helping engineers evaluate designs before physical testing. Codex-12B is the 12-billion-parameter code-generating model evaluated by Chen et al. (2021); HumanEval refers to the original 164-problem Python benchmark introduced in the same study to assess whether generated solutions work under specified conditions. The table then asks which part of each operation was delegated and what the evidence does—and does not—establish.

I, O, and C follow Table 2. Functional specifications and classifications are this article's analytical reconstructions; the cited sources establish the underlying operations and contexts. No absolute priority or representative long-run trend is inferred from these four cases. Conditions in the final column are proposed evidentiary checks, not experiments claimed to have been performed.

Table 4. Application to selected cases: functional reconstruction, classification, and evidentiary checks.

Case and Evidence	Function: I/O/C	Classification and Historical Role	Evidence That Would Change the Judgment
Automatic turret-lathe cycle Rosenberg (1963, pp. 428–431); §4.2.	I: workpiece and required geometry. O: machined component meeting the specified geometry. C: tools and initial setup specified; loading, maintenance, and inspection distinguished from cycle execution.	Intensive relative to the comparable machining operation, through reduced direct execution during the cycle. Recursive participation requires use in producing additional or improved machinery; repeated manufacture alone is insufficient.	Evidence that a relevant predecessor performed the same transformation with no greater human intervention would weaken the proposed intensive classification. Manual performance of the substantive shaping operation changes the delegation boundary. The source supports no universal priority claim.
A-0, 1951–1952 IEEE History Center (2024), milestone citation; §4.7.	I: specifications referring to available subroutines. O: linked and loadable program configuration. C: prepared library and computing environment; algorithm design and writing the routines remain human work.	Delegation of a bounded software-construction operation. Extensive status is not assigned because the cited record does not exhaust antecedents. Its role is to document software-producing operations before contemporary AI.	Evidence that the operator performed the linkage attributed to the system would revise this reconstruction. A functionally equivalent predecessor would defeat a first-execution claim. This record alone cannot establish increasing recursive depth.
SPICE, 1973 Nagel and Pederson (1973, §§I–III); §§4.7, 6.1.	I: circuit specification, component models, and analysis conditions. O: simulated electrical response. C: numerical and circuit-size limits; model validity; interpretation and physical validation kept separate.	Intensive relative to CANCER for documented modeling and analysis extensions. Simulation participates recursively in design evaluation: it performs an operation in developing circuits. The report supports a concrete comparison with manual analysis.	No added domain or condition would remove the intensive classification for that comparison. No use in development would remove recursive attribution in a particular project. A program genealogy alone does not establish a secular trend.
Codex/HumanEval, 2021 Chen et al. (2021, §§2–3); §6.4.	I: Python function signature and description. O: code satisfying declared tests. C: model version, sample allowance, test environment, and reliability criterion; human selection or correction recorded separately.	Documented, bounded generative performance. Extensive versus intensive status requires comparable antecedents. Passing benchmark tests does not by itself establish production deployment, project-level recursion, or productivity gains.	Independent tests or a predeclared reliability threshold may narrow the attributed capability. Substantive human correction changes the execution boundary. Successful cases cannot retrospectively define the entire admissible input population.

5. One Industrial Revolution, Multiple Technological Discontinuities

Throughout this paper, industrialization denotes the sustained transformation of production, technology, organization, and economic structure conventionally associated with development from the late eighteenth century onward (Landes 1969; Allen 2009). The long-run interpretation follows capabilities across its later technological phases; it does not assign a common start date to all countries or redefine every earlier invention as industrialization. Pre-industrial delegation is an antecedent against which the later dynamics must be compared, not evidence that the historical distinction disappears.

The proposed single Industrial Revolution is not a single homogeneous phase. Sectoral employment can undergo major reallocations, energy constraints can change, and new technological configurations can become dominant within a longer connected transformation. Table 1 documents the distinct criteria underlying these articulations. From the functional perspective, the question is whether successive changes extend and reconfigure an inherited field of technical execution, rather than whether employment or energy indicators remain continuous. The existence of multiple phases is therefore acknowledged; their relation to a common process remains to be argued from the evidence.

Simple continuity in one variable would not be sufficient. Energy consumption, scientific knowledge, market integration, and state capacity can all display long-run continuity across conventional period boundaries. Nor does mere chronological inheritance establish historical unity: almost every technology uses some earlier knowledge or artifact. Cumulative dependence is material here only when removing a capability generated in an earlier phase would make the later technological system infeasible in its historical form or would substantially change its feasible scale, complexity, performance, or resource requirements. Such dependence may be embodied in infrastructures, production capabilities, standards, organized knowledge, or institutional routines.

The historical cases provide more specific grounds than chronological succession. Rosenberg's (1963) account connects machining capabilities developed in one industry to the production requirements of others: later manufacturing draws on identifiable ways of shaping and reproducing components. Electrification and mass production then reorganize inherited machinery, power supply, and production arrangements rather than merely replacing one named invention with another (Hughes 1983; Hounshell 1984; David 1990). In this reconstruction, the persistent element is not an unchanged technology but the reuse and alteration of capabilities within new productive configurations.

Computing makes the connection particularly visible in both directions. It depends on a material electrical and manufacturing base (Ceruzzi 2003), while APT applies computed instructions to machine tools and computer-aided design feeds calculation back into the production of hardware (Ross 1978; Computer History Museum n.d.). SPICE provides a documented operation within that feedback: simulated responses help evaluate circuit designs that would be impractical to analyze by hand (Nagel and Pederson 1973). This is not simply the coexistence of machinery and software; a delegated operation in one domain enters the development process of another. Contemporary code generation supplies a further bounded case, examined in Section 6.4, but is not needed to establish that such relations predate AI.

These connections give historical content to the unity hypothesis: later changes in the set of technically executable functions draw on capabilities inherited and reused in subsequent development. One historically bounded contrast is the transition from mechanized machine production to programmable computing. This marks a genuine discontinuity in technical architecture and in how operations are specified (Ceruzzi 2003); viewed through delegated functions, however, machine tools had made precision shaping available for the production of further machinery (Rosenberg 1963), APT later linked computed instructions to machine–tool execution (Ross 1978), and SPICE performed calculations used to evaluate electronic-circuit designs (Nagel and Pederson 1973). The functional relationship lies in the documented reuse and extension of operations participating in subsequent technical development, not in the persistence of identical devices or a uniform rate of change. Taken together, these distinct historical connections support interpreting a major technological discontinuity as a transformation within a continuing industrial process, but do not establish the unity of every industrial trajectory. This interpretation neither displaces periodizations based on energy, employment, or techno-economic paradigms nor shows them to be in-

correct. The proposed interpretation would lose support wherever the apparent inherited capabilities proved incidental or the alleged delegated operations did not in fact enter subsequent development.

The common process is not mechanization alone, nor information processing alone. The proposed historical unity rests primarily on two forms of continuity across technological discontinuities: the continuing reallocation of executable functions between humans and technical systems, and the cumulative dependence of successive technological systems on capabilities generated within earlier industrial development.

The single-process hypothesis can therefore be stated through two primary conditions. First, functional continuity: successive phases continue to change which functions are technically executable by human and technical components. Second, cumulative continuity: later phases depend materially, in the counterfactual sense just specified, on capabilities generated in earlier industrial development. A third pattern—increasing recursive depth—is treated separately as an important but independently testable long-run characteristic: delegated functions appear to play an increasingly extensive and consequential role in producing subsequent technical capabilities, particularly after the emergence of programmable computing.

The two continuities are conditions to investigate, not a sufficient proof obtained from the terminology. The interpretation loses support where an alleged inherited capability proves merely incidental, where functional comparison cannot be sustained at consistent resolution, or where the proposed connections fail to extend beyond isolated examples. Section 7.7 distinguishes these checks from reclassification of individual functions and from the separate claim of increasing recursive depth. The hypothesis does not require either uninterrupted frontier expansion or a predetermined endpoint.

6. Computing, Digital Technology, and Artificial Intelligence

6.1. Computing as Flexibility and Recursion

The digital phase is particularly important because general-purpose computing combines high functional flexibility with recursive participation in technological development. Software weakens the correspondence between physical architecture and performed function. Once a general-purpose computational substrate exists, additional functional configurations can often be created, copied, modified, and combined without constructing a new specialized physical machine for each function. The records of automatic programming and circuit simulation in Section 4.7 identify concrete operations behind this claim, rather than treating programmability itself as proof of a long-run trend (IEEE History Center 2024; Nagel and Pederson 1973).

This does not make software immaterial. Programs depend on processors, memory, storage, networks, energy, fabrication systems, and increasingly specialized accelerators. The point is narrower: the material requirements of functional reconfiguration change. Compilers, libraries, operating systems, databases, development environments, simulation systems, and design tools accumulate as reusable layers. Functional execution becomes increasingly compositional. Histories of computing and electronic design document this material–software relation; the energy and infrastructure requirements of its contemporary AI applications remain observable constraints (Ceruzzi 2003; Wang et al. 2009; International Energy Agency 2026).

6.2. Networks and Distributed Functional Systems

Digital networks do not automatically move the frontier. Connectivity is not delegation. Networks nevertheless allow delegated functions to be combined across distributed systems: sensing can be separated from processing; storage from computation; software services can

expose functions to other software; remote outputs can affect physical systems elsewhere. The relevant technical unit increasingly becomes a networked functional configuration rather than an isolated machine. This can amplify flexibility, reuse, and recursive effects without making connectivity itself a frontier event. Their historical development also depended on negotiated architectures, institutions, and standards rather than on technical logic alone (Abbate 1999).

6.3. Machine Learning

Machine learning can loosen, for selected functions, the relationship between explicit rule specification and functional behavior. Architectures, training procedures, objectives, data, evaluation, and deployment conditions remain designed, but the complete operative mapping from inputs to outputs need not be explicitly hand-coded. This can broaden technical executability where explicit rule construction is difficult. The historical classification remains function-specific: many machine-learning advances are intensive expansions of existing technical functions, while extensive claims require evidence that a comparable input–output transformation was not previously technically demonstrated. The following case uses a dated code-generation study to make the input, output, performance conditions, and limits explicit (Chen et al. 2021).

6.4. AI-Assisted Software Development: A Contemporary Application

Code-generation systems provide a contemporary case for applying the functional-delegability framework. Automatic programming and compilers already delegated parts of the transformation from specifications to executable programs (§4.7). Chen et al. (2021) provide a later, different record: Codex was evaluated on HumanEval, a set of 164 Python programming problems with function descriptions and tests. Their 12-billion-parameter model achieved 28.8% pass@1. This is evidence of bounded performance under a declared evaluation procedure, not reliable execution of every problem or autonomy over a software project.

For that case, I comprises a function signature and description, O is code satisfying the stated tests, and C includes the model version, sampling allowance, testing environment, and accepted error. Tests are evidence about correctness within their coverage, not a guarantee of all possible behavior. Table 4 consequently leaves the extensive/intensive judgment dependent on a comparison with earlier program-generation systems at the same resolution. The existence of a benchmark result establishes neither a historical first nor recursive participation in a particular development project.

Observed work outcomes supply a different kind of evidence. Peng et al. (2023) randomized access to GitHub Copilot, using the pre-release version available during May–June 2022, for 95 developers implementing a JavaScript HTTP server; among participants completing the task, the assisted group's mean completion time was 55.8% lower. Becker et al. (2025), studying 16 experienced open-source developers on 246 tasks in familiar repositories with early-2025 tools, estimated a 19% increase in completion time when AI use was allowed. Different populations, tasks, tools, and designs preclude treating these results as directly comparable estimates of one universal effect. They do show why a capability demonstration cannot substitute for investigation of its productive use.

Material and economic conditions remain relevant at this level: generating code requires computing infrastructure, while choosing, checking, and integrating it consumes human effort. Recursive participation must therefore be identified in a concrete workflow: which generated or evaluated artifact enters a subsequent technical system, and which development operation changes as a result? The cited studies document bounded tech-

nical performance and work outcomes. A general transformation of service employment, autonomous innovation, or a secular increase in recursive depth is not established by them.

The infrastructure supporting AI also has environmental consequences. The International Energy Agency (2026, §1.3) identifies constraints in grid access, electrical equipment, advanced-chip packaging, and memory supply that can delay expansion. These are limits on implementation, not a refutation of a demonstrated function. Xiao et al. (2025) model the energy–water–climate implications of US AI-server deployment over 2024–2030, showing their sensitivity to utilization, cooling, electricity supply, and location. The study identifies opportunities for mitigation as well as constraints on achieving net-zero goals; its scenarios are not observed global outcomes. Together, these sources indicate why the scale and location of deployment matter environmentally. The IEA estimates worldwide data-center electricity consumption at 485 TWh for 2025 and projects approximately 950 TWh by 2030; consumption by AI-focused data centers is projected to triple over the same period (International Energy Agency 2026, Executive Summary). Neither total data-center electricity use nor its projected growth can be attributed wholly to AI, and neither source measures the footprint of a particular programming task. The general implication for diffusion is considered in Section 7.6.

7. Discussion

7.1. Circularity and Periodization

The single-Industrial-Revolution interpretation must not be produced by definition. Industrialization is identified through established historical evidence concerning sustained transformations in production, energy, organization, technology, and economic structure. The functional-delegability framework is then applied to ask whether a distinct continuity appears across conventional technological periods. The interpretation succeeds only if the proposed functional and cumulative continuities survive comparative historical testing. The proposition concerning increasing recursive depth remains independently testable.

7.2. Is Recursion Merely Cumulative Innovation?

All technological development is cumulative to some degree. The narrower claim made here is that functions already delegated to technical systems perform substantive parts of technological development, and may do so at increasing depth. Section 3.5 requires both a constitutive development operation and a material difference to the process if that operation is removed; usefulness alone does not suffice. SPICE's role in circuit evaluation gives this criterion a documented application. Establishing a historical increase in depth would additionally require comparable development processes from different periods, including cases in which human execution remained decisive.

7.3. Measurement and Historical Classification

The framework is comparative and classificatory rather than a scalar index. Table 4 uses a common record: the specified I/O/C, the antecedent, the documented operation, the classification, and evidence that would require its revision. Missing reliability, intervention, or predecessor information remains a limitation rather than being supplied by assumption. Extensive claims require a search for comparable predecessors; intensive claims require an identifiable enlargement of domain or execution conditions. Flexibility concerns functions executable through a common architecture. Recursive depth concerns the complexity and role of delegated operations within development, not the general sophistication of the artifact. These dimensions need not move together.

7.4. Teleology and AI

The framework does not imply a sequence from physical to informational to cognitive delegation. Regulation and information processing have deep histories; physical and informational functions coexist throughout industrialization; digital systems repeatedly return to physical execution through control, numerical manufacturing, robotics, logistics, and infrastructure. AI is therefore not the endpoint built into the framework. It is a contemporary test case whose long-run significance remains open. This emphasis on coexistence also accords with histories that redirect attention from celebrated novelty toward the continued use, maintenance, recombination, and afterlives of older technologies (Edgerton 2007; Weber 2024).

The corresponding question for service employment is not whether an entire sector becomes delegable. Service categories contain different functions, and shifts in sectoral employment can reflect changing demand and relative productivity as well as automation (Clark 1940, chap. V; Acemoglu and Restrepo 2019b). AI may therefore accompany redistribution within services, across existing sectors, or toward activities not yet prominent; this framework alone predicts none of these destinations. Whether such reallocations have an endpoint is also distinct from whether additional functions remain technically delegable. The article assumes neither an endless succession of sectors nor AI as a final stage of industrial transformation. A qualitatively different economic configuration, possibly altering the conditions under which scarcity is experienced, is another open hypothesis: a movement of the technical frontier alone cannot establish that such a transformation will occur.

7.5. Relationship to Marx, Beniger, Mokyr, GPTs, and Task Approaches

The framework is intentionally complementary to established traditions. From Marx ([1867] 1976) it inherits sensitivity to productive functions embodied in machinery, while abstracting from ownership and class relations to isolate technical executability. From Beniger it takes seriously the embedding of information and control inside industrial history, while treating control as one functional domain rather than the organizing principle. From Mokyr it accepts the importance of sustained technological change and useful knowledge, without offering a competing explanation of their origins. From GPT theory it borrows attention to complementarity and pervasiveness while changing the unit of analysis from technologies to functions. From task-based economics it shares the human–machine boundary while analytically separating technical possibility from economic task allocation. Table 1 adds an explicit comparison with employment-based, energy-based, and techno-economic interpretations: these explain structural shifts, enabling constraints, and systemic reconfigurations, respectively. Functional delegability addresses the executable operations and historical connections that may cross their phase boundaries; it does not replace their explanatory objects.

7.6. Delegability Is Not Destiny

A movement of the frontier does not establish that a function should be delegated, will be profitable to delegate, or will displace human labor. Actual delegation depends on costs, reliability, complementary investment, organizational capability, regulation, infrastructure, accountability, and social acceptance. At the organizational level, dependency restructuring distinguishes technological capability from enacted changes in activities and in the requirement relations among them (Sternieri 2026). This is a complementary account of implementation, not evidence for the historical unity proposed here. Demonstration under selected conditions does not imply robust organizational delegation, and successful delegation need not imply full automation.

Resource availability and environmental impacts can constrain the scale, location, and duration of deployment through physical limits, costs, infrastructure choices, and social

or regulatory responses. The material conditions that permit expansion and the impacts that expansion produces must both be examined (Wrigley 2010, chap. 9; International Energy Agency 2026; Xiao et al. 2025). A technically feasible practice may persist while imposing unsustainable burdens; conversely, mitigation can alter its prospects for wider use. Thus sustainability is neither guaranteed by technical success nor an automatic historical filter that prevents harmful diffusion. The AI case in Section 6.4 illustrates this distinction: more efficient computing or cooling can reduce impacts per operation, while aggregate consequences still depend on deployment scale, energy supply, and location. Environmental limits may therefore shape the historical realization of delegability without changing the criterion for its technical demonstration.

7.7. What Would Falsify the Framework or the Historical Interpretation?

Three judgments require different evidence. First, a functional classification can be challenged at fixed I/O/C. A documented equivalent predecessor defeats an extensive classification; failure to achieve the declared output or reliability without unreported human execution defeats the attributed level of delegability. For the HumanEval example, an independent assessment would fix the model, prompts, sample allowance, tests, and acceptance threshold before evaluation. It would record failures and human corrections, rather than select successful outputs to redefine the admissible domain afterward. Such a procedure could restrict the capability attributed to that configuration; it would not establish that code generation is impossible in general. For example, the reported 28.8% pass@1 for Codex on HumanEval (Chen et al. 2021) rules out treating that configuration as consistently satisfying the benchmark tests with a single generated sample, while preserving the more limited observation that some specified functions were technically executed. Persistent disagreement between independent classifications after these specifications were fixed would weaken the framework's reproducibility.

Second, the long-run interpretation requires evidence about the links between cases. For a proposed dependency, historical records must identify the inherited capability, its actual use in later development, and the consequence of replacing or removing it. Design records, production records, contemporary technical reports, and documented alternative methods are pertinent evidence. A record of comparable development without the claimed capability, with no material change in feasibility or resource requirements, would weaken that dependency claim. Nagel and Pederson's (1973) account of the difficulty of manual circuit analysis supports one such link; contrary contemporary evidence could qualify it. No technology need be independent of all preceding knowledge for a proposed link to fail. Repeated failure of the specific links across the transitions used to support the argument, or equally informative reconstructions in which they were incidental, would weaken the unifying interpretation. A gap in surviving records limits support but does not by itself prove historical independence.

Third, increasing recursive depth requires comparison of development operations across periods at consistent resolution. Researchers would record which design, generation, testing, evaluation, and production operations were technically executed, how complex those operations were, and what role humans retained. Evidence of no broadening, reduced delegated participation, or merely supportive rather than constitutive operations would count against a general increase. The presence of one simulator or code generator is insufficient. Table 4 demonstrates bounded classifications and their limits; these wider comparisons remain a research requirement, not a test reported as completed. Failure of the depth claim would not by itself reject either the classification scheme or every possible argument for a single industrial process.

7.8. Research Agenda

The framework suggests a comparative research program. First, does recursive depth measurably increase during industrialization relative to technologically sophisticated pre-industrial societies? Second, how consistently can independent historians classify extensive and intensive frontier movements using F = (I, O, C)? Third, does increasing functional flexibility correlate with faster or broader frontier movement in programmable technical systems? Fourth, what proportion of functions involved in designing and producing new technologies become technically delegated over time? Fifth, under what organizational conditions does technical delegability become actual delegation and productivity improvement?

Sixth, how do shifts in the functional-delegability frontier relate to historical transformations identified through sectoral employment, energy systems, and productivity, and what temporal or causal relationships, if any, can be established between these different dimensions? Comparative research should examine whether changes precede, accompany, or follow one another and which organizational, infrastructural, and environmental conditions mediate any observed relationships, without presuming that changes in technical delegability necessarily produce sectoral, energy, or productivity transformations. For contemporary AI, further studies could examine how energy, water, and material requirements affect sustainable implementation and how deeply delegated functions contribute to developing subsequent technical capabilities.

8. Conclusions

The history of industrialization is conventionally organized around its most visible technological discontinuities: mechanization, steam, electricity, mass production, electronics, computing, digital networks, and artificial intelligence. This paper has proposed a different variable: the functional-delegability frontier, the historically demonstrated boundary of functions technically executable by technical systems under specified conditions.

The framework defines functions independently of mechanism as F = (I, O, C). It distinguishes extensive expansion, in which a previously non-demonstrated function enters technical executability, from intensive expansion, in which the domain or operating conditions of an already delegable function broaden. It separates both from functional flexibility, the capacity of a common architecture to realize heterogeneous functions without substantial physical reconstruction. Finally, it introduces recursive expansion and recursive depth to capture the extent to which delegated functions materially participate in producing further technical capabilities.

The historical contrasts establish why no simple pre-industrial/industrial binary can be based on the existence of delegation. They also document more specific industrial connections: cross-sectoral reuse of machining capabilities, automatic operations in software construction, and computation returning to the design and production of hardware. These connections support the plausibility of functional and cumulative continuity across technological phases. They do not quantify a universal increase in recursive depth or establish the full genealogy of industrialization.

The comparison with employment-based, energy-based, and techno-economic interpretations makes the periodization claim more precise. Profound changes in sectoral composition, energy constraints, or technological organization need not be denied for functional reallocation to be interpreted as part of one still-unfolding Industrial Revolution. The hypothesis rests on the documented connections among capabilities, not simply on choosing a different variable. Increasing recursive depth is proposed as an additional long-run characteristic rather than a necessary condition of historical unity.

The article offers a historically supported hypothesis, not an established conclusion that displaces alternative periodizations. Broader comparative work is needed across

pre-industrial societies, industrial trajectories, and technological domains. Concrete contemporary records make the framework applicable to AI, but do not settle future occupational destinations, the duration of industrial transformation, or the sustainability of its implementation.

Its purpose is narrower: to make the changing set of technically executable functions visible as a historical variable in its own right. From that perspective, the question raised by contemporary AI is not simply whether another Industrial Revolution has begun. It is how AI changes a functional frontier whose movement—and whose increasing participation in producing its own subsequent technical possibilities—may be one of the enduring characteristics of industrialization.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Acknowledgments: During manuscript revision, ChatGPT (OpenAI, GPT-5.6 Sol) was used to support source retrieval and comparison, drafting, structural revision, and language editing. It was not treated as a source of evidence. Responsibility for the source interpretations, analysis, and final text rests with the author.

Conflicts of Interest: The author is employed by Energee3. The author declares that the company was not involved in the study design, analysis, interpretation of the results, writing of the manuscript, or the decision to submit it for publication.

References

Abbate, Janet. 1999. *Inventing the Internet*. Cambridge: MIT Press.

Acemoglu, Daron, and Pascual Restrepo. 2019a. Artificial Intelligence, Automation, and Work. In *The Economics of Artificial Intelligence: An Agenda*. Edited by Ajay Agrawal, Joshua Gans and Avi Goldfarb. Chicago: University of Chicago Press, pp. 197–236. [CrossRef]

Acemoglu, Daron, and Pascual Restrepo. 2019b. Automation and New Tasks: How Technology Displaces and Reinstates Labor. *Journal of Economic Perspectives* 33: 3–30. [CrossRef]

Allen, Robert C. 2009. *The British Industrial Revolution in Global Perspective*. Cambridge: Cambridge University Press.

Arthur, W. Brian. 2009. *The Nature of Technology: What It Is and How It Evolves*. New York: Free Press.

Autor, David H., Frank Levy, and Richard J. Murnane. 2003. The Skill Content of Recent Technological Change: An Empirical Exploration. *Quarterly Journal of Economics* 118: 1279–333. [CrossRef]

Becker, Joel, Nate Rush, Elizabeth Barnes, and David Rein. 2025. Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity. *arXiv* arXiv:2507.09089v2. [CrossRef]

Beniger, James R. 1986. *The Control Revolution: Technological and Economic Origins of the Information Society*. Cambridge: Harvard University Press.

Bissell, Christopher. 2009. A History of Automatic Control. In *Springer Handbook of Automation*. Edited by Shimon Y. Nof. Berlin/Heidelberg: Springer, pp. 53–69. [CrossRef]

Bresnahan, Timothy F., and Manuel Trajtenberg. 1995. General Purpose Technologies "Engines of Growth"? *Journal of Econometrics* 65: 83–108. [CrossRef]

Ceruzzi, Paul E. 2003. *A History of Modern Computing*, 2nd ed. Cambridge: MIT Press.

Chandler, Alfred D., Jr. 1977. *The Visible Hand: The Managerial Revolution in American Business*. Cambridge: Belknap Press.

Chen, Mark, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, and et al. 2021. Evaluating Large Language Models Trained on Code. *arXiv* arXiv:2107.03374v2. [CrossRef]

Clark, Colin. 1940. *The Conditions of Economic Progress*. London: Macmillan.

Computer History Museum. n.d. 1966: Computer Aided Design Tools Developed for ICs. The Silicon Engine. Available online: https://www.computerhistory.org/siliconengine/computer-aided-design-tools-developed-for-ics/ (accessed on 26 August 2026).

David, Paul A. 1990. The Dynamo and the Computer: An Historical Perspective on the Modern Productivity Paradox. *American Economic Review* 80: 355–61.
Edgerton, David. 2007. *The Shock of the Old: Technology and Global History Since 1900*. Oxford: Oxford University Press.
Garcia, Chris. 2018. Artifact Adopted! Preserving the Legend of J. M. Jacquard. *Computer History Museum*, December 21. Available online: https://computerhistory.org/blog/artifact-adopted-preserving-the-legend-of-j-m-jacquard/ (accessed on 26 August 2026).
Hill, Donald R. 1991. Arabic Mechanical Engineering: Survey of the Historical Sources. *Arabic Sciences and Philosophy* 1: 167–86. [CrossRef]
Hounshell, David A. 1984. *From the American System to Mass Production, 1800–1932: The Development of Manufacturing Technology in the United States*. Baltimore: Johns Hopkins University Press.
Hughes, Thomas P. 1983. *Networks of Power: Electrification in Western Society, 1880–1930*. Baltimore: Johns Hopkins University Press.
IEEE History Center. 2024. A-0 Compiler and Initial Development of Automatic Programming, 1951–1952. IEEE Milestone. Available online: https://ethw.org/Milestones:A-0_Compiler_and_Initial_Development_of_Automatic_Programming,_1951-1952 (accessed on 26 August 2026).
International Energy Agency. 2026. *Key Questions on Energy and AI*. Paris: IEA. Available online: https://www.iea.org/reports/key-questions-on-energy-and-ai (accessed on 13 October 2026).
Kline, Ronald R. 2015. *The Cybernetics Moment: Or Why We Call Our Age the Information Age*. Baltimore: Johns Hopkins University Press.
Landes, David S. 1969. *The Unbound Prometheus: Technological Change and Industrial Development in Western Europe from 1750 to the Present*. Cambridge: Cambridge University Press.
Mahoney, James, and Dietrich Rueschemeyer, eds. 2003. *Comparative Historical Analysis in the Social Sciences*. Cambridge: Cambridge University Press.
Marx, Karl. 1976. *Capital: A Critique of Political Economy*. Translated by Ben Fowkes. London: Penguin, Vol. 1. First published 1867.
Mokyr, Joel. 1990. *The Lever of Riches: Technological Creativity and Economic Progress*. New York: Oxford University Press.
Mokyr, Joel. 2002. *The Gifts of Athena: Historical Origins of the Knowledge Economy*. Princeton: Princeton University Press.
Mokyr, Joel. 2005. The Intellectual Origins of Modern Economic Growth. *Journal of Economic History* 65: 285–351. [CrossRef]
Myers, Oscar. 1952. Common Control Telephone Switching Systems. *Bell System Technical Journal* 31: 1086–120. [CrossRef]
Nagel, Laurence W., and Donald O. Pederson. 1973. *SPICE (Simulation Program with Integrated Circuit Emphasis)*. Memorandum UCB/ERL M382. Berkeley: University of California, April 12. Available online: https://www2.eecs.berkeley.edu/Pubs/TechRpts/1973/Archive/ERL-m-382.pdf (accessed on 13 October 2026).
Needham, Joseph. 1965. *Science and Civilisation in China*. Vol. 4, Physics and Physical Technology, Part 2, Mechanical Engineering. With the Research Assistance of Wang Ling. Cambridge: Cambridge University Press.
Noble, David F. 1984. *Forces of Production: A Social History of Industrial Automation*. New York: Alfred A. Knopf.
Peng, Sida, Eirini Kalliamvakou, Peter Cihon, and Mert Demirer. 2023. The Impact of AI on Developer Productivity: Evidence from GitHub Copilot. *arXiv* arXiv:2302.06590. [CrossRef]
Perez, Carlota. 2002. *Technological Revolutions and Financial Capital: The Dynamics of Bubbles and Golden Ages*. Cheltenham: Edward Elgar.
Perez, Carlota. 2009. *Technological Revolutions and Techno-Economic Paradigms*. Working Papers in Technology Governance and Economic Dynamics 20. Tallinn: Tallinn University of Technology. Available online: https://hum.ttu.ee/wp/paper20.pdf (accessed on 13 October 2026).
Perez, Carlota, and Tamsin Murray Leach. 2022. *Technological Revolutions: Which Ones, How Many and Why It Matters: A Neo-Schumpeterian View*. BEYOND4.0, Deliverable D7.1. London: UCL Institute for Innovation and Public Purpose.
Randell, Brian. 1994. The Origins of Computer Programming. *IEEE Annals of the History of Computing* 16: 6–14. [CrossRef]
Rosenberg, Nathan. 1963. Technological Change in the Machine Tool Industry, 1840–1910. *Journal of Economic History* 23: 414–43. [CrossRef]
Rosenberg, Nathan. 1982. *Inside the Black Box: Technology and Economics*. Cambridge: Cambridge University Press.
Ross, Douglas T. 1978. Origins of the APT Language for Automatically Programmed Tools. *ACM SIGPLAN Notices* 13: 61–99. [CrossRef]
Sternieri, Armando. 2026. Dependency Restructuring: How Technological Enactment Changes Organizational Requirement Relations. *Information* 17: 1015. [CrossRef]
Toomer, G. J. 2015. Ctesibius, Inventor, fl. 270 BCE. In *Oxford Classical Dictionary*. Oxford: Oxford University Press. [CrossRef]
Truesdell, Leon E. 1965. *The Development of Punch Card Tabulation in the Bureau of the Census, 1890–1940: With Outlines of Actual Tabulation Programs*; Washington, DC: U.S. Government Printing Office.
Wang, Laung-Terng, Yao-Wen Chang, and Kwang-Ting Cheng, eds. 2009. *Electronic Design Automation: Synthesis, Verification, and Test*. Burlington: Morgan Kaufmann/Elsevier.
Weber, Heike. 2024. Beyond Innovation and Use, or Why We Must Follow Technologies through Time. *Histories* 4: 51–61. [CrossRef]
Wrigley, E. A. 2010. *Energy and the English Industrial Revolution*. Cambridge: Cambridge University Press. [CrossRef]

Xiao, Tianqi, Francesco Fuso Nerini, H. Damon Matthews, Massimo Tavoni, and Fengqi You. 2025. Environmental Impact and Net-Zero Pathways for Sustainable Artificial Intelligence Servers in the USA. *Nature Sustainability* 8: 1541–53. [CrossRef]

Yates, JoAnne. 1989. *Control Through Communication: The Rise of System in American Management*. Baltimore: Johns Hopkins University Press.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.