



WORKING PAPER

16/2020

**Drivers of automation and consequences for jobs  
in engineering services: an agent-based modelling  
approach**

Hildegunn Kyvik Nordås and Franziska Klügl  
*Economics*

ISSN 1403-0586

# Drivers of automation and consequences for jobs in engineering services: an agent-based modelling approach

Hildegunn Kyvik Nordås and Franziska Klügl

December 23, 2020

## Abstract

This paper studies the uptake of AI-driven automation and its impact on employment, using a dynamic agent-based model (ABM). It simulates the adoption of automation software as well as job destruction and job creation in its wake. There are two types of agents: manufacturing firms and engineering services firms. The agents choose between two business models: consulting or automated software. From the engineering firms' point of view, the model exhibits static economies of scale in the software model and dynamic (learning by doing) economies of scale in the consultancy model. From the manufacturing firms' point of view, switching to the software model requires restructuring of production and there are network effects in switching. The ABM matches engineering and manufacturing agents and derives employment of engineers and the tasks they perform, i.e. consultancy, software development, software maintenance, or employment in manufacturing. Policy parameters influencing the results are occupational licensing and protection of intellectual property rights. We find that the uptake of software is gradual; slow in the first few years and then accelerates. Software is fully adopted after about 18 years in the base line run. The adoption rate is slower the higher the license fee for software, while the adoption rate is faster the higher the mark-up rate of consultancy. Employment of engineers shifts from consultancy to software development and to new jobs in manufacturing. Spells of unemployment may occur, if skilled jobs creation in manufacturing is slow. Finally, the model generates boom and bust cycles in the software sector.

JEL: C51, C61, J44, L84, O33

**Keywords**— Technology Uptake, Employment, Automation, Economic Modelling, Agent-Based Simulation

## 1 Introduction

Due to recent advances in algorithms and technology based on Artificial Intelligence (AI), intelligent automation systems are rapidly moving into the workplace. Nevertheless, the adoption of technology is gradual, often with long lags between innovation and adoption. A survey on firms' use of AI released by Statistics Sweden in November 2020, for example, finds that only 5.4% of the firms surveyed use AI.<sup>1</sup> Against this backdrop, it is clear that to assess the impact of AI on the future of work, one first needs to understand what determines the uptake of AI in firms. This paper studies the uptake of AI-driven

---

<sup>1</sup>See [https://www.scb.se/contentassets/4d9059ef459e407ba1aa71683fcbd807/uf0301\\_2019a01\\_br\\_xftbr2001.pdf](https://www.scb.se/contentassets/4d9059ef459e407ba1aa71683fcbd807/uf0301_2019a01_br_xftbr2001.pdf).

automation as a market interaction between developers and users of software and ready-to-use systems. It contributes to the literature in three major ways.

First, it models the adoption of technology rather than innovation. Studies on the adoption of AI are few and far between despite the observed long lags between innovation and adoption. The driving forces that we analyze are economic and institutional, notably uncertainties about user costs and benefits of the new technology, the need for skills upgrading, and regulatory incentives or disincentives. Tracing out the AI adoption trajectory is important e.g. for informing skills- and labor market policy.

Second, the study focuses on AI-adoption in services, noting that AI-enabled automation is also on its march into the services sectors.<sup>2</sup> Indeed, the Swedish AI adoption survey found that services sectors that produce and use ICT intensively have the highest AI-adoption rate in the economy. Furthermore, a recent EU enterprise survey on the use of technologies based on AI found that about 60% of firms buy software or ready-to-use systems from external services suppliers.<sup>3</sup> We model the uptake of AI-enabled software as a market interaction between engineering firms and manufacturers where engineering firms develop software or ready-to-use systems for use in manufacturing. The software and services nexus is empirically important, but under-researched as most existing studies focus on the impact of robotics in manufacturing. One important reason for this is that while data on robot use is readily available, data on AI-enabled software use is not.

This leads to our third major contribution, which is to develop an agent based model (ABM) to study the joint adoption of AI in services and manufacturing. ABMs are particularly suitable for dynamic processes where outcomes are uncertain and agents interact (Dawid 2006). Furthermore, it is apposite when the future is likely to be qualitatively different from the past such as during technological transitions (Farmer and Foley 2009). Our ABM captures the interactions between the agents and the environment in which they operate and generates important insights on the trajectory of AI adoption. Notably, the model generates the boom and bust cycles often observed during the early stages of technology adoption.

Our model has two types of agents, engineering firms and manufacturing firms; and two business models, which we label consultancy and software respectively. Consultancy is the traditional business model where engineering firms deploy consultants to clients, working with them on-site and face-to-face to solve technical problems. Recent developments in AI techniques enable engineering firms to develop intelligent systems embedding Automated Reasoning or Machine Learning based on easily accessible open source tools such as Keras<sup>4</sup>. Experienced engineers can use their understanding of their particular service domain and increasingly available data sets to create software solutions that automate tasks that were previously handled by consultants. Gathering this experience is modelled as learning-by-doing and represents dynamic economies of scale. Ultimately, the engineering firm may devote staff to R&D to harness accumulated experience into automation software which may be licensed to manufacturing clients. Examples of such software may be intelligent systems for computer-assisted design (CAD), computer-aided process planning (CAPP), computer-aided manufacturing (CAM) and computer numerical controller (CNC). Manufacturers buy such software through licensing agreements, paying a license fee, or they may opt for cloud-based software-as-a-service, paying an annual subscription rate.<sup>5</sup> Manufacturers decide whether to license software or stick to the consultancy model based on the expected costs and benefits of doing so. The benefits are uncertain at the time of the decision. Our analysis shows that it is hesitance on the part of manufacturers that holds back the uptake of AI-enabled software.

The rest of the paper is organized as follows: Section two briefly discusses related research while section three develops a conceptual framework that captures the interaction between engineering firms, their clients and the environment in which they operate. The framework is coded into a dynamic ABM in section four. Section five presents the simulation results, while section six summarizes and concludes.

---

<sup>2</sup>See for instance Baldwin and Forslid 2020.

<sup>3</sup>See <https://ec.europa.eu/digital-single-market/en/news/european-enterprise-survey-use-technologies-based-artificial-intelligence>.

<sup>4</sup>*keras.io*

<sup>5</sup>See X. V. Wang, Givchchi, and L. Wang 2017 for a discussion of use of CAx in manufacturing.

## 2 Relations to previous work

During the last decade Agent-based Modelling and Simulation has become an established microsimulation approach in social science, ecology and for modeling complex systems in general Klügl and Bazzan 2012. The underlying metaphor of such a model is a Multi-Agent System that means a set of interacting agents – that can be basically seen as situated intelligent, autonomous actors Wooldridge 2009. A model captures agents’ decision making in their individual environmental context. During simulation, overall dynamics are generated. Consequently, agent-based simulation is particularly apt for modeling endeavor which heterogeneous agents, with transient dynamics and without the necessity of an equilibrium-based model. Also, in economical research, agent-based simulation becomes more and more relevant Gallegati and Richiardi 2009; *Agent-Based Modelling in Economics* 2016.

Our paper is the first interdisciplinary study of the uptake of technology and its impact on jobs drawing on insights from economics and computer science, using ABM. There is, however a growing body of literature in the social sciences, notably economics, on automation and the future of jobs. The most common approach is to break jobs down to tasks (Autor, Levy, and Murnane 2003), identify tasks that can be automated using existing technology and make predictions about the future of work from analysing the task content of different occupations. This approach has been criticized for exaggerating job destruction from technological change. First, early papers did not take into account that the bundle of tasks that constitute a job or an occupation is not necessarily separable Arntz, Gregory, and Zierahn 2016. Second, the fact that technology also creates jobs is largely overlooked in the most alarming studies David 2015.

Acemoglu and Restrepo 2018 addressed both job creation and lags in technology adoption, in principle. They framed technical progress as a race between “man and machine” in a process where technical advances allow tasks previously performed by workers to be automated. At the same time innovation creates new tasks for which man has a comparative advantage over machines. The framework consists of a static model that focuses on automation and its impact on employment and wages and a dynamic extension that focuses on the direction of R&D towards automation or innovation. It includes two scenarios, one in which allocation of tasks between man and machine is constrained by technology, the other where it is not. In the former case firms would always find it profitable to automate as soon as it is technologically possible. In the latter scenario, firms operate well within technological limits and would automate, innovate or neither, depending on the relative cost of labour and machines. The paper does, however not explore this second scenario any further. A possible reason is that it is hard to explain that gainful economic activities are not exploited. Nevertheless, economic history describes long lags between invention, innovation, and diffusion of technology as well as different rates of adoption between firms Rosenberg 1972; Brynjolfsson and Hitt 2000; Fagerberg 2004. This paper explores the non-technology-constrained scenario using an ABM which features interaction between agents that do not have perfect information and foresight.

A recent paper that applies ABM in a context similar to Acemoglu and Restrepo 2018 is Neves, Campos, and Silva 2019. They studied the relationship between innovation and employment within the framework of a task-based model where tasks differ in terms of automation potential.<sup>6</sup> They trace out the creative destruction of innovation and automation in a model with one type of agent (firms). Our model differs from theirs by introducing the interaction between two types of agents, adding network effects and a policy dimension. Furthermore while *ibid.* studied the impact on employment of product innovation versus process innovation, we are interested in how intermediate services input developed by engineering firms shape manufacturing technology adoption as well as what functions engineers perform.

---

<sup>6</sup>Neves, Campos, and Silva 2019 appear, however, to be unaware of the Acemoglu and Restrepo 2018 paper, suggesting limited knowledge spillovers across disciplines in this area.

## 3 The model

### 3.1 Intuition

We propose a dynamic model consisting of two types of agents: engineering firms and their manufacturing clients. Manufacturers produce final goods according to a production function which combines production workers employed by the manufacturer and services inputs sourced from engineering firms. We distinguish two types of relationships between the engineering firms and the manufacturer: consultancy and software.

The consultancy model involves engineers working with the client, on-site, face to face, to solve problems and perform a set of tasks. The problems are client-specific and the ability to solve them rests with the consultant. The engineering firm and the manufacturer enter a contract which specifies the tasks the consultant is to perform and the payment, which is an annual fee per consultant.

In the software model the engineering firm establishes an R&D department where it develops software that automates services. The R&D activity requires a given number of engineers and their salaries constitute a fixed cost which the engineering firm recuperates through the licensing of the resulting software. Once developed, the software can be licensed to an unlimited number of clients. On the client side, licensing of software requires organizational adjustments. Therefore, to make full use of the software, the manufacturer hires engineers to operate the software and streamline it into the production process.

Each engineering firm offers its unique variety of the service, and thus distinguishes its product from competitors. Such product differentiation implies that the engineering firms may charge customers a premium and mark up their price over marginal cost. In the case of occupational licensing, engineers have exclusive rights to perform a predefined set of tasks. Furthermore, they may limit the number of licenses issued and thereby charge a higher mark-up.

Manufacturers are heterogeneous in terms of size and productivity. Productivity is a measure of how effectively the firm transform inputs into outputs. Thus, the more productive firms use less engineering services per unit of output. Switching business model from relying on external consultants to using software involves restructuring of production for a seamless interface between fabrication and the software. This requires upgrading of machinery and skills, creating jobs for engineers to manage the interface between the software and machinery, supervise production workers and govern the licensing contract with the engineering firm.<sup>7</sup>

The dynamics of the model consist of learning by doing on the part of engineers working on problem-solving in manufacturing firms and network effects in the adoption of software. The ABM model matches engineering consultants and manufacturers as well as software licensors to manufacturing licensees in each period. At the end of each period the model updates the level of experience gathered by each consultant, adjusts the number of surviving agents in each business model, new agents may enter, and finally the model allocates workers across firms and activities.

### 3.2 Formal model

Manufacturers, indexed  $i$ , are heterogeneous in terms of productivity denoted  $\theta_i$ , which follows a Pareto distribution. The probability density function of the Pareto distribution is given by  $g(\theta) = k(\theta_{min})^k(\theta)^{-(k+1)}$  where  $\theta_{min}$  is the scale parameter, which we set to unity, and  $k$  is the shape parameter, which we tentatively set to 2.2.<sup>8</sup> The corresponding cumulative density function is  $1 - \left(\frac{\theta_{min}}{\theta}\right)^k$ . The manufacturers produce final output, denoted  $Y$  using production workers indexed  $l$  and engineering services. Total costs for the

---

<sup>7</sup>There is ample evidence that ICT and AI complement skills in the workplace. See for instance Berman, Bound, and Machin 1998; Autor, Levy, and Murnane 2003; Brynjolfsson, Rock, and Syverson 2019; Bessen et al. 2018

<sup>8</sup>This is close to empirical estimates of the shape parameter of productivity distribution from firm level data Feenstra 2018. A shape parameter larger than 2 ensures that the variance of the distribution can be identified.

consultancy and software models respectively are:

$$TC_{i,c} = \left[ \frac{w_l}{\alpha} + \frac{\varphi w_s}{\beta_c \theta_i} \right] Y \quad (1)$$

$$E[TC_{i,softw}] = \frac{A}{\theta_i} w_l^{1-\beta} w_s^\beta Y + \delta + \gamma \quad (2)$$

$TC$  represents total cost of production. The two business models are indexed  $c$  and  $softw$  respectively. In both cases we apply constant elasticity of substitution production- and cost functions. In the consultancy model we use the extreme case of a Leontief specification where production factors are perfect complements, while in the software model we apply the Cobb-Douglas functional form. These particular functional forms are not critical for the results, but serve to distinguish between more and less flexible technologies in the two business models.

Variables and parameters:  $\alpha$  represents the production worker intensity of production while  $\beta_c$  depicts the consultancy input intensity of production. Wage rates for production workers and engineers are denoted by  $w_l$  and  $w_s$  respectively, while the mark-up rate that engineering firms obtain for their consultancy services is  $\varphi$ . A scale parameter  $A$ , the license fee for software,  $\delta$  and a stochastic element  $\gamma$  are additional parameters in the cost function for manufacturers that opt for the software model. The stochastic element  $\gamma$  is normally distributed  $\gamma \sim \mathcal{N}(0, \sigma^2)$ . Manufacturers will be in the market for software if the expected cost of switching to software is lower than continuing with the consultancy model. Figure 1 illustrates the two cost functions where the horizontal axis represents manufacturing firms' productivity and the vertical axis total cost. Clearly, software represents the lowest cost for high-productivity firms, while consultancy is the better option for low-productivity manufacturers.<sup>9</sup>

There are network effects related to the switch to the software model as adopters reorganize production, including relations to suppliers and customers around the software. We capture this by modelling the scale parameter  $A$  to be a declining function of the number of firms that have switched to software. The network effect works with one period lag.

$$A_t = \frac{A_{t-1}}{n_{softw,t-1}^\mu} \quad (3)$$

where  $0 < \mu < 1$ . Demand for engineering consultancy services from each manufacturing firm choosing that model is given by:

$$C_i = \frac{Y}{\beta_c \theta_i} \quad (4)$$

Manufacturers that have switched to the software model will seek to employ engineers according to the demand function:

$$S_i = \frac{A}{\theta_i} \left[ \frac{\beta}{1-\beta} \frac{w_l}{w_s} \right]^\beta Y \quad (5)$$

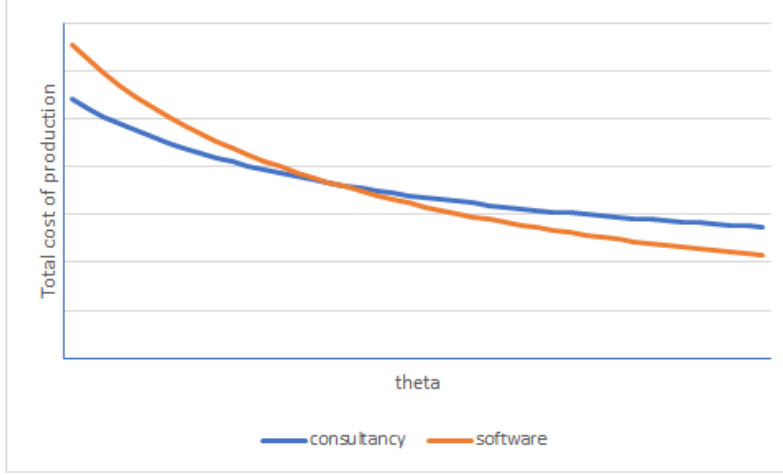
Engineering firms, indexed over  $j$  hire engineers which are deployed to client firms on a contractual basis in the consultancy model. The contract covers one period and its value varies across clients, depending on their size and productivity as indicated in the demand function, equation 4. The engineering firms incur wage costs only and they sell consultancy services with a mark-up factor of  $\varphi > 1$ . The consultancy revenue is thus  $\varphi w_s \sum C_i$ . We choose units such that one unit of consultancy services corresponds to the input of one full-time consultant for one period. Profits from the consultancy model are:

$$\pi_{c,j} = (\varphi - 1) \sum C_i \quad (6)$$

---

<sup>9</sup>Production workers and the market for final output determine the size of the economy and relative scarcity of engineers, but are not the focus of the analysis.

Figure 1: Total cost comparison, consultancy and software



In the software model, engineering firms establish an R&D department and divert  $S_F/\lambda$  engineers to staff it. The R&D department uses AI-based platforms and available data sets, applying machine learning (ML) to create software that accomplishes the tasks that are otherwise done by consultants. Engineers accumulate experience from working with clients, and harness this experience into software that is automating the consultants' work.

It is assumed that a minimum number of inexperienced engineers are needed to successfully develop the software. So, experience accumulated over years of working with clients is an advantage when developing software, assuming that it helps to identify appropriate learning architectures or to formalize knowledge for automated reasoning. We model this by introducing the experience of the engineer, denoted  $\lambda$  in the cost of developing the software. The total cost of switching for the engineering firm is the wage costs for the engineers working in the R&D department and the foregone profits from no longer deploying them to clients as consultants. Revenue in the software model will be the licence fee  $\delta$  times the number of manufacturers that license the software from company  $j$ ;  $n_{softw,j}\delta$ . Expected profits from the software model is thus:

$$E[\pi_{softw,j}] = E[n_{softw,j}]\delta - \frac{w_s S_F}{\lambda} \varphi \quad (7)$$

The engineering firm knows the cost of developing software, but at the point of decision whether to develop it, the number of clients that will take up the software is unknown. The engineering firm does, however observe the productivity of the manufacturers and thus can estimate, how many of them are sufficiently productive to gain from switching to software. Engineering firms base their decision to develop software on expectations about how many clients they may capture from the mass of manufacturers that are sufficiently productive to benefit from switching to the software model.

After software is available, manufacturers that decided to switch their business model to software, randomly select engineering companies that offer software. Random selection is weighted by experience of the software provider assuming that more experienced firm produce higher quality software. Since the marginal cost of servicing another client is zero, it is conceivable that one engineering firm could corner the market.

It is clear from equation 7 that profits from switching to the software model is lower the higher the mark-up factor  $\varphi$ , predicting that engineering firms operating in a less competitive market, for instance a market with occupational licensing, are less innovative than firms operating in a competitive market which limits the ability to charge a high price.<sup>10</sup> Engineering firms will develop the software if expected profits as defined in 7 is positive.

After the initial investment into software development, the traditional software lifecycle contains a number of periods with software maintenance. It is assumed that data-driven software is depreciating fast, thus we assume that software lasts  $T$  periods. Each period between its development and obsolescence a fraction  $\zeta$  of the number of engineers that are needed to develop the software, is then needed to maintain it. After  $T$  periods, the engineering firm needs to invest again into full software development.

Experience accumulates from working on-site and face to face with manufacturing clients. Furthermore, engineers gain more experience from working with the most productive manufacturers. An engineering firm  $j$ 's accumulated experience is thus a function of the productivity of the manufacturers it has worked with as follows:

$$\lambda_{j,t} = \int_0^t f(\theta_{ij}) d\theta \quad (8)$$

These eight equations, representing supply and demand for engineering services in two business models constitute the conceptual core of the ABM. The forces that drive the adoption of software are engineers' accumulated experience from working with clients and network effects from its adoption. What holds back the development of software is comfortable profits from the consultancy model on the part of the engineering firms, uncertainty about how many manufacturers will buy the software once the cost of developing it is sunk, and uncertainty about the gains from the switch to software on the part of manufacturers. These countervailing forces ensure a gradual adoption of software in the economy. The speed depends on the size of the economy, the endowment of production workers and engineers, the level and dispersion of productivity among manufacturing companies as well as policy-induced factors including occupational licensing and protection of intellectual property rights.

## 4 The ABM setup

The agents and their role and actions are presented in Table 1

The environment consists of supply of production workers and engineers, a set of exogenous parameters and decision rules as spelled out in the model presented in Section 3. The decisions by the agents and coordination of their activities are depicted in Figure 2.

The simulation runs through the following phases:

---

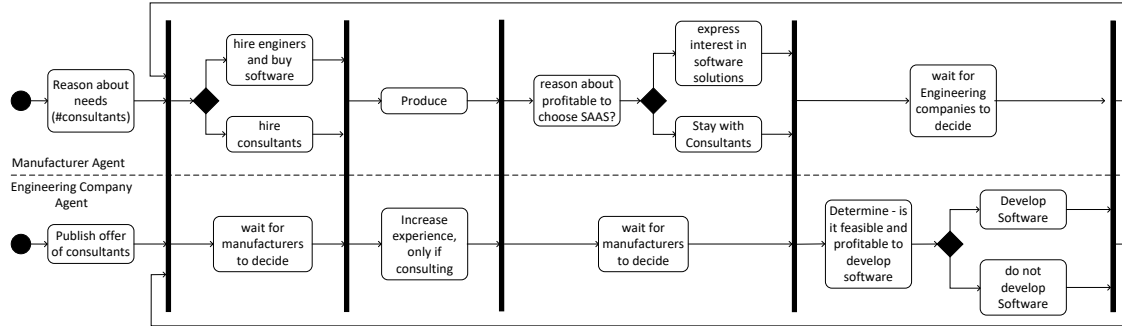
<sup>10</sup>International trade in engineering services would also limit the ability to charge a high mark-up and thus spur innovation.



Table 1: Agents

Agents	Status	Role and actions	
		Consultancy	software
Manufacturing firms	Active	Employ production workers Enter consultancy contract	Employ production workers License software Employ engineers Produce final output
		Produce final output	
Engineering firms	Active	Employ engineers Enter consultancy contract	Employ engineers Develop and maintain software License out software
Production workers	Passive	Work in manuf. firms	Work in manuf. firms
Engineers	Passive	Work in eng. firms	Work in any firm
Authorities	Passive	Occupational licensing, IPR protection	

Figure 2: Activities of the different agent types and their coordination.



Note: The black vertical lines form synchronization bars meaning that all agents need to have finished the activities before an individual agent can continue with the next activity after the bar.

- In phase 0 – during initialization –, manufacturing firms draw their productivity level from a Pareto distribution. Manufacturers hire production workers, which are matched to firms randomly, but the number of employees is proportional to the firms’ productivity. Engineering firms hire engineers, which are randomly matched to engineering firms.
- In phase 1 – first year – all firms adopt the consultancy business model. Engineering firms and manufacturers are matched randomly and manufacturers produce final output.
- At the end of phase 1, all active firms observe their profits. Engineering firms’ experience parameter is updated. Engineering firms next consider whether to develop software and automate their service. The decision is based on expected number of clients ready to switch to the software model, and the cost of developing the software. The cost is lower for the more experienced firms. Engineering firms observe the productivity of manufacturers. They expect to sell to a random subset of the manufacturers ready to switch. If expected profit from selling software is positive, engineering firms will establish an R&D department which will work on software development. Redundant engineers, if any, are laid off. Manufacturers decide whether to switch to the software model. The decision is conditioned on software being available as well there being engineers available on the market to hire

in the new jobs created during the switch to the software model. The most productive manufacturers are the first to switch to software. If expected profits from the software model is smaller than that for continuing with the consultancy model for all engineering firms, phase 1 is repeated and consulting engineers gain more experience for each repetition.

- In phase 2 at least one engineering firm has developed software and earns a positive profit from licensing it. In this phase the two business models coexist. Manufacturers having switched to software license software from a random supplier and hire engineers to integrate the software into the production process, other manufacturer continue hiring consultants. Manufacturers that do not license software and do not find consultants, do not produce output, all others do. Not all engineering firms developing software may be profitable. Making a loss from software development, causes engineering firms to immediately return to offering consultancy services.
- Manufacturers' cost of switching to software is adjusted by the network effect given by equation 3, that means with more and more manufacturers using software, it becomes cheaper to also use software. Software is maintained (bug fixes, new, minor features in small updates) at a cost  $\zeta S_F$  with  $0 < \zeta < 1$ . When a software has reached obsolescence, the engineering firm decides again whether in a changed market, it could generate profit when re-developing software. Experience of engineers working as consultants is updated. So, the situation is not static – due to co-adaptation, a dynamic process is ongoing. Phase 2 continues until all manufacturing firms have switched to software.
- In phase 3 all firms have switched to software. There is a churning of engineering firms as software becomes obsolete and new software is developed to replace it. At this stage, engineers no longer gain experience from working directly with clients, but more are employed to support the software usage at the manufactures. There is still some dynamic ongoing at the engineering firms, as manufacturers re-select software in each period – we do not assume commitment to a particular software product. That means producing software in market in which every manufacturer uses software, may also cause a loss, especially investing into new development.

Exogenous variables and parameters are summarized in Table 2

Table 2: Exogenous variables and parameters

Symbol	Description	Value in the baseline case
L	Number of production workers	3000
S	Number of engineers	1000
$w_l$	salary, production worker	1
$w_s$	salary, engineer	1.5
$\alpha$	production worker intensity, manufacturing, consultancy model	1
$\beta_c$	consultant intensity, manufacturing, consultancy model	1.5
$\beta$	engineer intensity, manufacturing, software model	0.2
$\theta_i$	productivity level, manufacturing firm $i$	Pareto distributed
$A_0$	scale parameter, manufacturing, software model	3
$\mu$	strength of network effects of using software	0.02
$\delta$	license fee, software	10
$\gamma$	stochastic switching cost, manufacturing	normally distributed
$\lambda_0$	initial experience, engineers	1
$\eta$	update factor for $\lambda$	0.1
$\varphi$	mark-up rate, consultancy	1.3
$S_F$	number of engineers needed to develop software	18
$\zeta$	Software maintenance cost relative to development cost	0.5

The ABM was implemented using the SeSAM platform<sup>11</sup> which is a fast prototyping environment for

<sup>11</sup>[www.simsesam.org](http://www.simsesam.org)

agent-based simulations providing an activity diagram-like way of implementing complex agent behavior.

## 5 Results

We start by running the simulations with baseline parameters as reported in table 2, including sensitivity analysis on the overall size of the economy and the ratio of production workers to engineers. We experiment next with policy relevant parameters: i) the mark-up rate, which is related to the strength of competition in the engineering services market and ii) the license fee, which is partly related to the strength of intellectual property rights protection and partly to the strength of competition in the market for software. Eventually, we want to explain what are the relevant factors influencing how fast intelligent automated solutions distribute in a market characterized by the parameters above as well as what is the actual impact and dynamics on the employment of highly qualified engineers.

### 5.1 Baseline

We start by simulating the baseline scenario<sup>12</sup>. As described, we start with a scenario where all firms are in the consultancy model. Firms next decide whether to switch to the software model and look for a supplier or customer for software. As Figure 3 indicates, a few manufacturing firms already switch to software in the second year. All engineering firms anticipate the market opportunity these firms constitute, and a large share of them decides to develop software. However, the customers are few, competition is fierce, and most early software developers fail. As a consequence, those failing firms give up to offer software<sup>13</sup>.

The uptake of software in manufacturing is gradual and about half of all manufacturers have switched to software after 11 years. The uptake does, however, accelerate after about a third of all manufacturers have switched, and levels off when about 90% of firms have switched. During the first decade of relatively slow uptake, there is a competitive fringe of engineering firms that develops software, fails and exits as indicated by the zigzagging of the blue line in the chart. After all manufacturers have switched to the software business model, about 80% of engineering firms offer software. There remains a competitive fringe of engineering firms that exit when a loss from software happens, when a new development is necessary, but too expensive or when simply not a sufficient number of software licenses were acquired. A start-ups seeks consultancy contracts, but realizes that demand for such services is close to zero and quickly starts to develop software as well.

What we see in our simulation shown in Figure 3 is a largely demand-driven adoption of automation software, and a boom and bust cycle in the automation software sector. The simulations thus predict a boom-bust cycle similar to the so-called dot.com bubble that we observed in the 1990s when adoption of ICT took off.<sup>14</sup>

For explaining this overall behavior, a look into the dynamics on the agent level is helpful. Figure 4 depicts the lifeline of two randomly selected engineering firm agents. They both start out as consultants and earn a positive profit. They both end up profitably licensing software, and they both have at least one unsuccessful attempt at switching to software. The first company has two spells of consultancy after a commercially successful software becomes obsolete, while the second company experiences only one such event.

Figure 5 shows the business model dynamics for all engineering firms over the complete simulation run. We observe that they have all entered the software model after two years, but only three are successful and continue in the third year to maintain and develop their software. As time passes, the dynamics turn

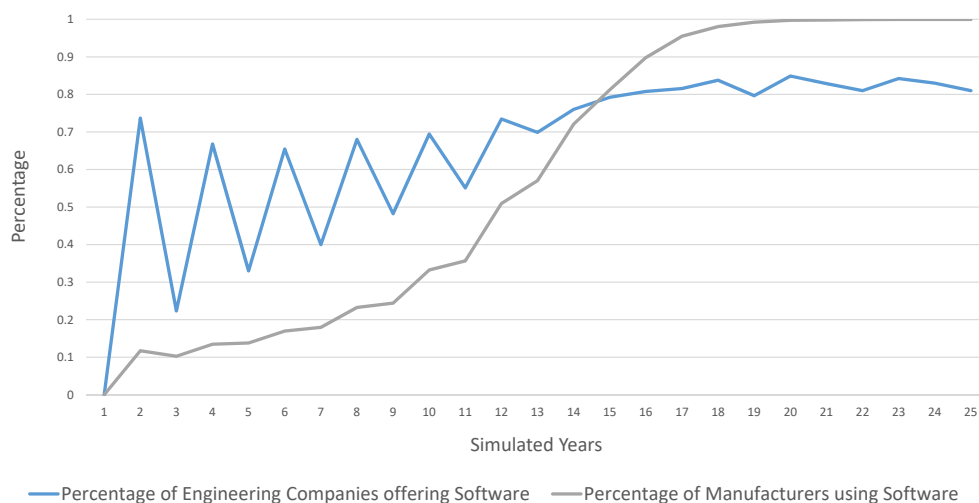
---

<sup>12</sup>We repeat every simulation 30 times. If not otherwise stated, diagrams show averaged values. Where suitable, we also give the standard distribution which is naturally higher in the transient phase 2 and low in phase 3

<sup>13</sup>Technically, this may be modelled as an exit of the firms that fail to sell the software they have developed, while start-up engineering firms use the consultancy model, or as a single firm switching between business models. The results are the same either way.

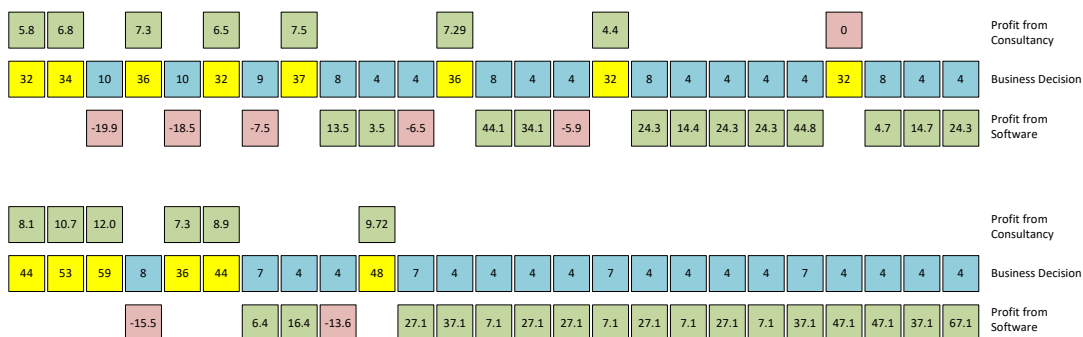
<sup>14</sup>See for instance Doms et al. 2004 for a study of the dot.com bubble in the US.

Figure 3: Percentage of companies having switched to the software model



Note: Baseline setting as described in Table 2.

Figure 4: Two examples of the lifeline of randomly selected engineering firms.

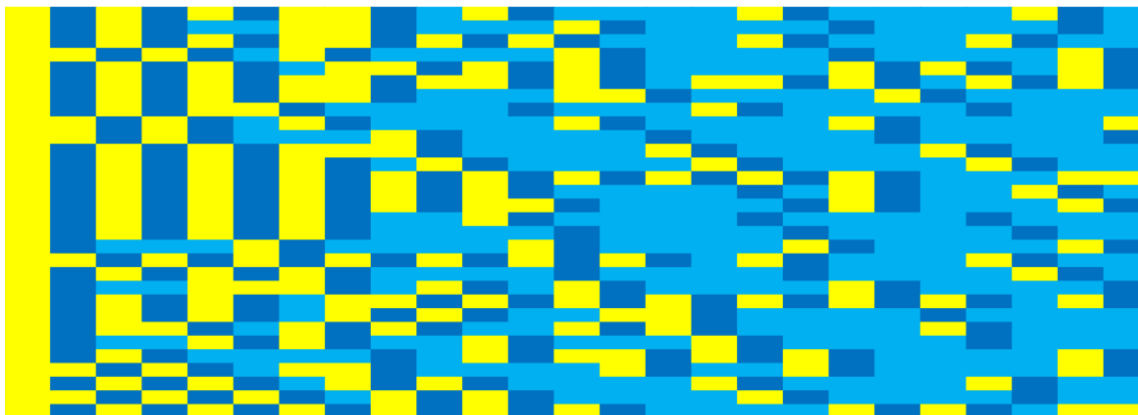


Note: The middle row shows the selected business model (consultancy in yellow, software in blue). Numbers in these cells denote the numbers of employed engineers in that year. The upper cells contain the profit the company made in cases of consultancy, the lower cells profit or loss when trying to sell software.

increasingly towards a shifting between developing new and maintaining existing software, but all firms experience occasional failures in the market for automated software.

In addition to the technology uptake, we also want to analyse dynamics of the engineer employment.

Figure 5: Business model dynamics of all engineering companies over 25 simulated years.



Note: Yellow means that the agent offers consulting, dark blue: it develops software, blue: it maintains software.

Figure 6 depicts the dynamic impact of technology adoption on employment of engineers. All engineers work as consultants in the first year. Consistent with the changing business model, they gradually move to the R&D department in the engineering firm where they develop and subsequently maintain software. Consultants that cannot find a job in the R&D department are laid off. Most of them find new jobs in manufacturing firms that have switched to software and are looking for engineers to fill new jobs created during the transition to a more sophisticated and skills-intensive production process. Finally, some of the laid off engineers do not find a new job immediately, and become unemployed. We notice that with substantial economies of scale in software development. The number of workers needed to develop and maintain software is relatively small. Our simulations thus predict that most of the changes in employment are from external consultants to engineers working in manufacturing.

An interesting parameter is  $\beta$ , the engineer intensity influencing how many engineers are needed to support complex software usage at the manufacturer.

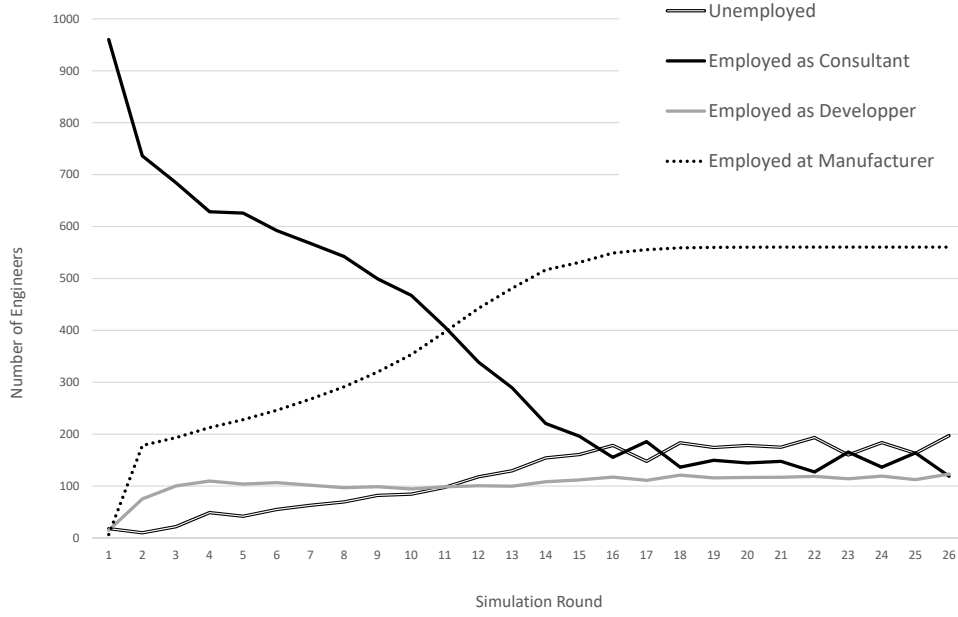
The unemployment rate among engineers following the transition to software depends crucially on the ratio of production workers to engineers in the labor market and the desired skills composition of employees in manufacturing firms that have switched business model. Sensitivity analyses depicted in Figure 7 shows that there will be full employment of engineers at the end of the transition period if *beta* is larger than about a quarter. Sensitivity analyses also shows that with a smaller engineer-to-production worker ratio there could also be shortage of engineers at lower levels of *beta*.

## 5.2 Experiments, the mark-up rate

The mark-up rate reflects the strength of competition in the market for consultant engineering services. High mark-up rates may stem from occupational licensing that gives licensed engineers exclusive rights to perform a defined set of engineering tasks, a small market closed to foreign competition, or simply a shortage of engineers for instance due to low education capacity for engineers or a limited number of engineering licenses issued.

From equations 2 and 7 we see that a high mark-up rate makes consultants relatively more expensive than software. On the other hand a higher mark-up rate yields higher profits for the engineers in the consultancy model. Thus, manufacturers are more likely to switch to software the higher the mark-up rate, while engineering firms are less likely to switch the higher the mark-up rate. It follows that if adoption of

Figure 6: Development of employment over simulation time. Overall number of engineers is 1000.



the software model is driven from the demand side, the adoption rate is faster the higher the mark-up. If on the other hand the uptake is driven by a supply-push, then we would expect it to be delayed for longer the higher the mark-up rate. Figure 8 clearly shows that this is a demand pull story.

Figure 9 shows employment of engineers by sector and activity after 25 years as a function of the mark-up rate. We first notice that employment of engineers in manufacturing is largely unaffected by the mark-up rate. After 25 years all manufacturers will have switched to the software model and pay engineers the going wage  $w_s$ , rather than the marked-up consultancy fee, so this is no surprise. Employment as software developer is also largely unaffected by the mark-up rate. Where we do see a significant difference is on the employment of consultants and the unemployment rate for engineers. The employment as consultant is actually very brittle as there is practically no market for consultancy services. Engineering firms that made a loss with software provision, try to re-establish with consultancy, yet there is hardly any demand for consultancy services and thus the profit from consultancy is zero. Such a company needs to revise its business model and moves to the software market and lays off the newly recruited engineers.

An important policy implication of the simulations is that the potentially harmful delay of the uptake of technology due to occupational licensing does not materialize in a demand-driven market. This conclusion holds when the mark-up rate is unrelated to the software license fee and thus, exclusive rights do not extend to software licensing. We now turn to an experiment where we let the license fee vary.

### 5.3 Experiments, the license fee

As Figure 10 indicates, the adoption rate of software is slower the higher the license fee. From equation 2 we observe that the cost of software is higher for manufacturers the higher is  $\delta$ , so there will be fewer takers of software the higher is  $\delta$ . This also results in weaker network effects, further slowing down the uptake over time (see equation 3). On the other hand, as depicted in equation 7 the revenue of the engineering

Figure 7: Influence of  $\beta$  on the employment structure.

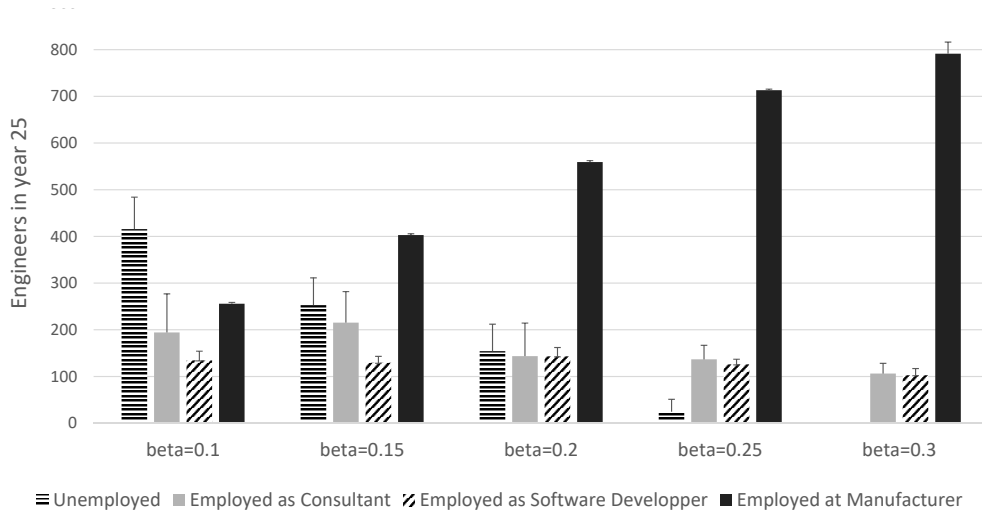
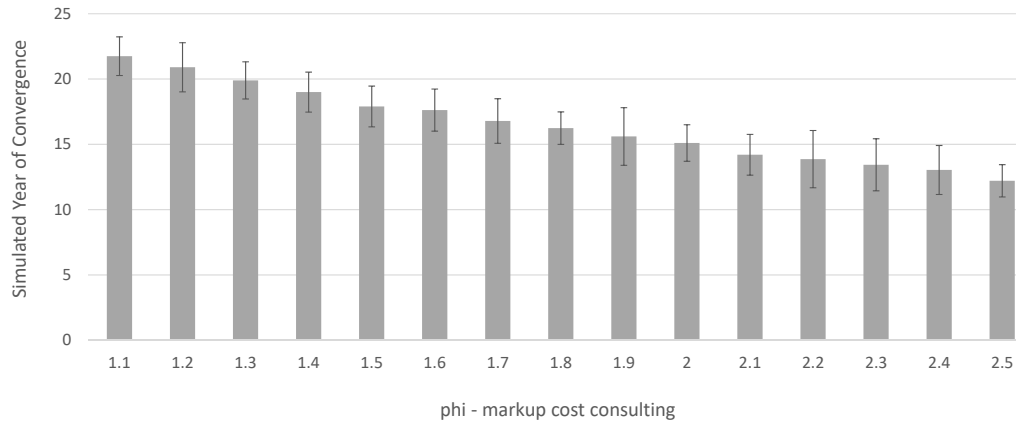


Figure 8: Mark-up rates and simulated year in which 100% manufacturers were using software

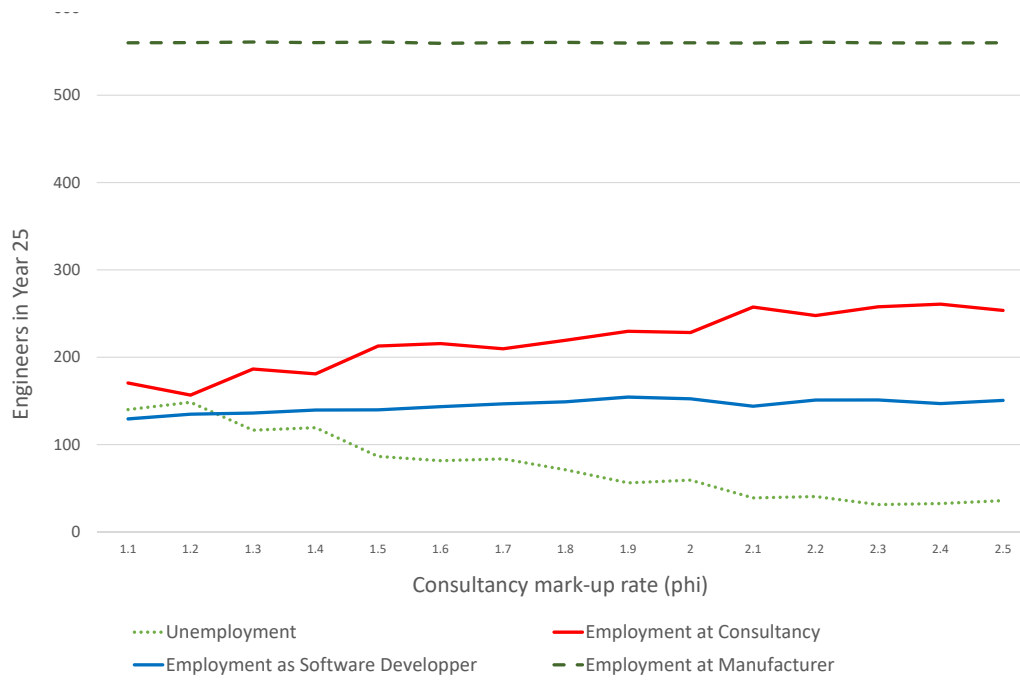


Note: Averaged over 30 simulation runs, standard deviation shown as error bars. In the case of  $\varphi = 1.1, 1.4$  and  $2.4$  two runs was omitted as not converged. With  $\varphi = 1.3, 1.6, 1.8$  and  $2.5$  convergence was not reached in one single run. We consider those cases as outliers with randomly generated manufacturers that are particularly small and omitted those from the analysis.

firms providing software is higher the higher is  $\delta$ , all else equal. Thus, the slower rate of transition to the software model stems from the demand side.

A convergence towards a situation in which all manufacturers use software happens also in scenarios with high licence fees. Longer simulation runs with  $\delta \geq 20$  confirm that the adoption rate eventually approaches 100%. For example, with  $\delta = 25$ , the population of firms converges into a stable situation with slightly less than 100% of manufacturers using software between years 50 and 60. In these simulations, two engineering firms are too small to start software development and try to hold on to the consultancy model

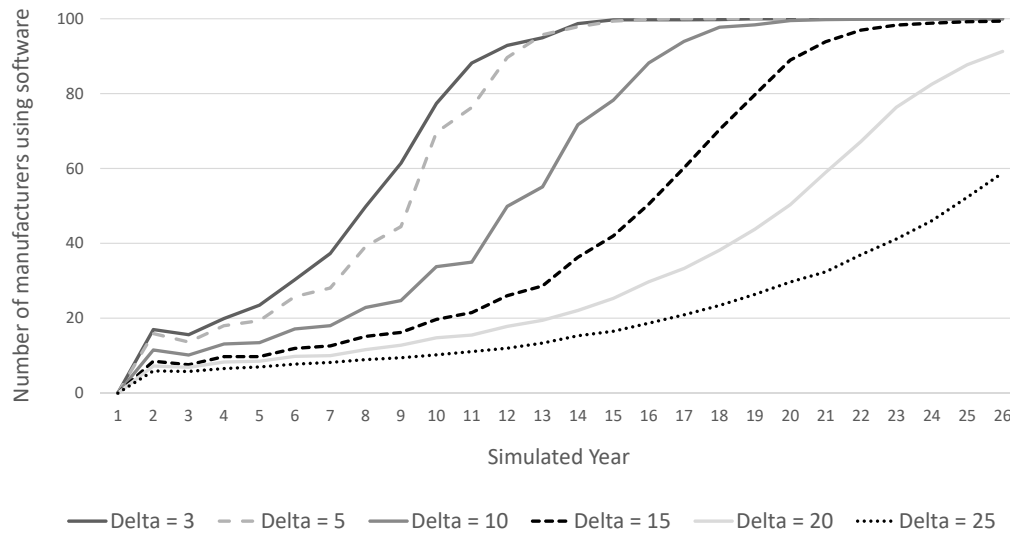
Figure 9: Employment in simulated year 25 depending on the consultancy mark-up rate. Averaged over 30 simulation runs.



before they exit.

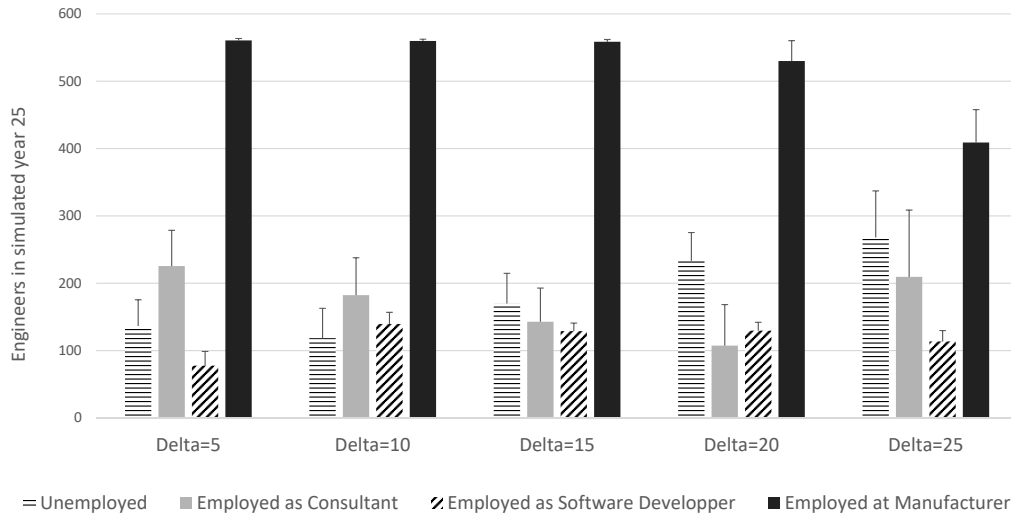


Figure 10: Number of manufacturers who use software over simulated time with different settings of the licence fee Delta.



A higher license fee also result in a higher rate of unemployment among engineers during and after the transition to software as illustrated in Figure 11. Consultancy jobs are lost, and job creation in the R&D department to develop and maintain software together with engineering jobs created in manufacturing is insufficient to absorb the idle consultants. However, sensitivity analyses with a higher  $\beta$ , substantially reduce or eliminate unemployment among engineers also when  $\delta$  is higher than in the baseline scenario.

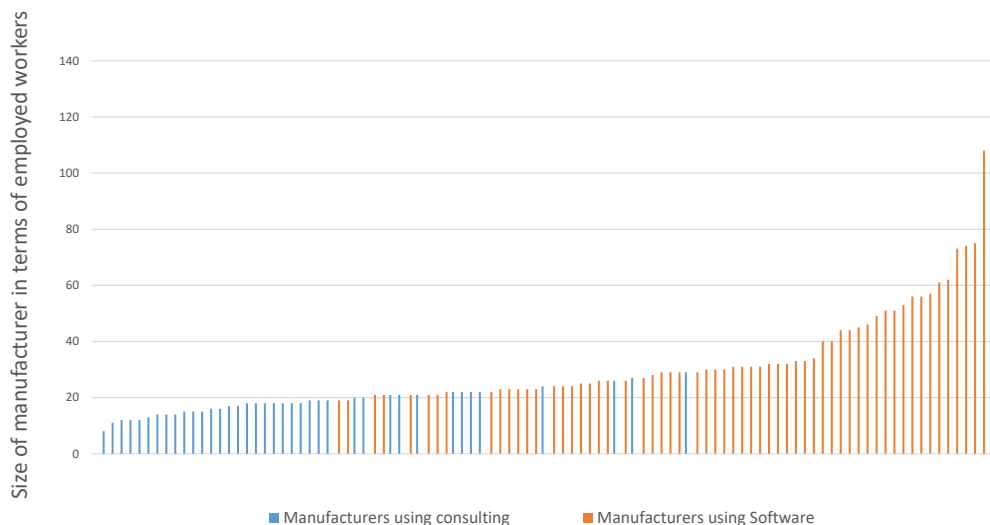
Figure 11: Employment of engineers in the simulated year 25.



Note: Average over 30 runs, the error bar shows the standard deviation between runs. Delta is the licence. All other parameter according to baseline scenario.

Finally, Figure 12 shows the number of manufacturing firms that take up software in a scenario where software is very expensive and the two business models co-exist also in the long run. It illustrates that the first adopters are the largest and most productive manufacturing firms. Further, since firms may have different risk assessments related to switching to software, there is a mix of software adopters and consultancy users in the middle range of firm size and productivity levels.

Figure 12: Illustration of the dependency between size of a manufacturer and its decision to use software.



Note: Delta = 50, in year 25. With  $\gamma$ , there is a random element in the decision making: for sizes between 19 and 29 employed workers, both decisions are observed, yet with corresponding tendencies.

## 6 Conclusion

Economic history documents that the adoption of technology is gradual with long delays. Furthermore, it is amply documented in the business literature that the adoption of new technology in firms requires organizational changes and new skills, which constitute significant switching costs for individual firms. Nevertheless, recent literature on AI and the future of jobs overlooks or abstracts from such switching costs and assume that AI-driven automation technology is adopted as soon as it is invented, with dramatic effects on jobs. To understand, predict and prepare for the labour market implications of AI on jobs a much better grasp on what drives the *adoption* of technology is needed. Our paper contributes to filling this gap, studying the adoption of AI-driven automation jointly in engineering services firms and their manufacturing customers.

Our simulations generate results that resonate with insights from economic history. First, AI-driven automation, like general purpose technologies before, is adopted gradually. It starts at a slow pace, and accelerates after reaching a critical mass of adoption. Second, switching costs on the user side is the most important factor holding back the adoption of new technology. Third, technology does indeed destruct jobs, but it also generates new jobs in the technology-using sectors. Finally, our simulations generate a boom and bust cycle on the supply side of the technology sector, which resembles what we have observed in the past, for instance during the dot.com bubble. This is not often observed in the literature and is thus an important contribution to new insight.

A policy implication of our findings is that innovation policy is not enough to foster technical progress. New technology is of little use if it is not adopted. We find that the early adopters are the largest and most productive manufacturing firms and that network effects of technology adoption is strong. Furthermore, we find that adoption of AI-driven automation is associated with demand for more skilled labor in using sectors. Policies aiming at fostering technical progress therefore needs to focus more on the user-side and

on education and skills to make sure that the potential users of new technology can find the skills needed to restructure production around the technology.

The importance of the demand-side also suggest that occupational licensing does not constitute a drag on technology adoption as long as at least one engineering firm offers software. However, if exclusive rights to offer a service extends to software automating the same service, the license fee is likely to be higher than in a competitive market, and the adoption rate may be substantially slowed down.

Finally, our results are relevant for other occupations and sectors. First, AI-enabled automation software in engineering is also relevant for the construction sector in a similar manner as in manufacturing. Second, other professionals such as architects and management consultants face similar technological changes as the ones simulated here for engineering. These professions have in common with engineering that they provide inputs to downstream sectors that may need to restructure production to benefit from the technology.

## References

- Acemoglu, Daron and Pascual Restrepo (2018). “The race between man and machine: Implications of technology for growth, factor shares, and employment”. In: *American Economic Review* 108.6, pp. 1488–1542.
- Agent-Based Modelling in Economics* (2016). John Wiley and Sons.
- Arntz, Melanie, Terry Gregory, and Ulrich Zierahn (2016). “The risk of automation for jobs in OECD countries”. In:
- Autor, David H, Frank Levy, and Richard J Murnane (2003). “The skill content of recent technological change: An empirical exploration”. In: *The Quarterly journal of economics* 118.4, pp. 1279–1333.
- Baldwin, Richard and Rikard Forslid (2020). *Globotics and development: When manufacturing is jobless and services are tradable*. Tech. rep. National Bureau of Economic Research.
- Berman, Eli, John Bound, and Stephen Machin (1998). “Implications of skill-biased technological change: international evidence”. In: *The quarterly journal of economics* 113.4, pp. 1245–1279.
- Bessen, James E et al. (2018). “The Business of AI Startups”. In: *Boston Univ. School of Law, Law and Economics Research Paper* 18-28.
- Brynjolfsson, Erik and Lorin M Hitt (2000). “Beyond computation: Information technology, organizational transformation and business performance”. In: *Journal of Economic perspectives* 14.4, pp. 23–48.
- Brynjolfsson, Erik, Daniel Rock, and Chad Syverson (2019). “Artificial Intelligence and the Modern Productivity Paradox”. In: *The Economics of Artificial Intelligence: An Agenda*, p. 23.
- David, HJJOEP (2015). “Why are there still so many jobs? The history and future of workplace automation”. In: *Journal of economic perspectives* 29.3, pp. 3–30.
- Dawid, Herbert (2006). “Agent-based models of innovation and technological change”. In: *Handbook of computational economics* 2, pp. 1235–1272.
- Doms, Mark et al. (2004). “The boom and the bust in information technology investment”. In: *Economic Review-Federal Reserve Bank of San Francisco*, pp. 19–34.
- Fagerberg, Jan (2004). “Innovation: A guide to the literature”. In: Georgia Institute of Technology.
- Farmer, J Doyné and Duncan Foley (2009). “The economy needs agent-based modelling”. In: *Nature* 460.7256, pp. 685–686.
- Feenstra, Robert C (2018). “Restoring the product variety and pro-competitive gains from trade with heterogeneous firms and bounded productivity”. In: *Journal of International Economics* 110, pp. 16–27.

- Gallegati, Mauro and Matteo G. Richiardi (2009). “Agent Based Models in Economics and Complexity”. In: *Encyclopedia of Complexity and Systems Science*. Ed. by Robert A. Meyers. New York, NY: Springer New York, pp. 200–224.
- Klügl, Franziska and Ana L. C. Bazzan (Sept. 2012). “Agent-Based Modeling and Simulation”. In: *AI Magazine* 33.3, p. 29. DOI: 10.1609/aimag.v33i3.2425. URL: <https://ojs.aaai.org/index.php/aimagazine/article/view/2425>.
- Neves, Fábio, Pedro Campos, and Sandra Silva (2019). “Innovation and Employment: An Agent-Based Approach”. In: *Journal of Artificial Societies and Social Simulation* 22.1.
- Rosenberg, Nathan (1972). “Factors affecting the diffusion of technology”. In: *Explorations in economic history* 10.1, p. 3.
- Wang, Xi Vincent, Mohammad Givehchi, and Lihui Wang (2017). “Manufacturing system on the cloud: a case study on cloud-based process planning”. In: *Procedia CIRP* 63, pp. 39–45.
- Wooldridge, Michael (2009). *An Introduction to Multiagent Systems*. second. John Wiley and Sons.