Automated Stem Cell Production by Bio-Inspired Control

László Monostori[a,b]*, Balázs Cs. Csáji[a], Péter Egri[a], Krisztián B. Kis[a], József Váncza[a,b], Jelena Ochs[d], Sven Jung[d], Niels König[d], Simon Pieske[c], Stephan Wein[c], Robert Schmitt[c,d], Christian Brecher[c,d]

[a] *Centre of Excellence in Production Informatics and Control, Institute for Computer Science and Control, Eötvös Loránd Research Network, Budapest, Hungary*
[b] *Department of Manufacturing Science and Engineering, Budapest University of Technology and Economics, Budapest, Hungary*
[c] *Laboratory for Machine Tools and Production Engineering (WZL), RWTH Aachen University, Aachen, Germany*
[d] *Fraunhofer Institute for Production Technology, Aachen, Germany*

* Corresponding author. Tel.: +36 1 279 6159; fax: +36 1 4667 503. *E-mail address:* laszlo.monostori@sztaki.hu

Abstract

The potential in treating chronic and life-threatening diseases by stem cell therapies can greatly be exploited via the efficient automation of stem cell production. Working with living material though poses severe challenges to automation. Recently, production platforms has been developed and tested worldwide with the aim to increase the reproducibility, quality and throughput of the process, to minimize human errors, and to reduce costs of production. A distinctive feature of this domain is the symbiotic co-existence and co-evolution of the technical, information and communication, as well as biological ingredients in production structures. A challenging way to overcome the issues of automated production is the use of biologically inspired control algorithms. In the paper an approach is described which combines digital, agent-based simulation and reinforcement learning for this purpose. The modelling of the cell growth behaviour, which is an important prerequisite of the simulation, is also introduced, together with an appropriate model fitting procedure. The applicability of the proposed approach is demonstrated by the results of a comprehensive investigation.

Keywords: stem cell production; digital simulation; machine learning; reinforcement learning; policy gradient; biological transformation

Introduction

Stem cell based therapies belong not only to the future but already to the present of regenerative medicine. Their full potential in treating chronic and life-threatening diseases can though be realized only if both clinical testing and treatment are supported by the efficient production of stem cells [1], [2]. Production is essentially the cultivation of living samples under controlled conditions. *Automation* here is also the key to efficiency, quality and cost-effectiveness. However, in contrast to traditional manufacturing processes, automating the production of stem cells has a number of severe challenges: (1) the inherent diversity of the products themselves (i.e., the stem cells), (2) their varying growth rates and, consequently, processing times, (3) the need for regular check-ups, observations and continuous process adaptation, which altogether call for (4) human-involvement and a mixed-initiative production control scheme. Hence, stem cells are typically produced with significant human involvement using adaptive protocols that take the growth behaviour of the biological material into account. The goal of this work was to devise novel methods for the automated production of stem cells so as to increase the reproducibility, quality and throughput of the process, to minimize human errors, and, last but not least, to reduce costs.

As stated in [3], "increased understanding of the underlying biological processes and their interaction with approaches to manufacturing technology will be needed to create the step change required for the next generation of scalable precision production systems capable of more than replicating and incrementally improving the performance of the human operator". The research reported here attempted a step in the above direction as far as it introduced a biologically inspired scheme – *reinforcement learning* (RL) – for controlling the process of stem cell production. The paper is an extended version of a former conference publication [4], presenting here a more detailed analysis and more comprehensive experimental results.

However, it is hard to find similar challenges in the traditional manufacturing processes. A rare exception is the wafer manufacturing process in the semiconductor industry, presented briefly in Section "Related works in production engineering". Section "Automated stem cell culture" discusses the main challenges of fully automatic stem cell production, and presents systems developed at the Laboratory for Machine Tools and Production Engineering (WZL), RWTH Aachen University and at the Fraunhofer Institute for Production Technology, Aachen. Here we introduce also models of the cell growth process along with methods of fitting the models to reality. An *agent-based simulation* model of the selected automated stem cell production system and the developed control concept are detailed in Section "Simulation of the stem cell production". Next, Section "Biologically inspired control of stem cell production" summarizes the first computational experimental results achieved by a new controller, which was generated by reinforcement learning. Finally, general conclusions are drawn.

Related works in production engineering

Production engineering in some traditional industrial domains has already accumulated experience and provided results whose transfer to the production of biologically materials are worth considering. Primarily, experimental production planning and control (PPC) solutions in the *semiconductor* industry are relevant, due to the following characteristics of the front-end (i.e., wafer) fabrication process. (1) The processing of wafers takes relatively long time (10-20 weeks) which is interleaved with many quality control steps; (2) the process is non-linear, products return to the same resources time and again; (3) processing times are stochastic; (4) there are tight temporal constraints due to the risk of contamination; (5) in-process buffer sizes are limited; as well as (6) order and machines statuses are available any time. The main key performance indicators (KPIs) of wafer production are to maximize resource utilization, and, simultaneously, to minimize throughput time (for more details, see [5], [6]).

Production in such a complex, dynamically changing system burdened by both product and process related uncertainties can only be scheduled and controlled by some *dispatching logic* which adapts to the actual situation at hand and decides in *real-time* but only on the *short term* what and where to do. These, usually, rule-based methods consider the routing of products, the availability of resources, the due dates, estimated processing and waiting times as well as rush orders, to name but the most important factors. However, setting the parameters and weightings of the rules, as

well as inferring the impact of their interplay can hardly be accomplished in a way which would warrant even feasibility, let alone quasi-optimality with respect to the KPIs. At the same time, advanced discrete-event *simulation* methods can capture even the fine-grained structure and detailed behaviour of such a production system.

Hence, recently, a new solution approach has evolved for controlling highly complex and dynamic semiconductor production systems which is based on learning from interactions with a detailed simulation model. In a reinforcement learning (RL) framework (see details in Section "Simulation of the stem cell production"), control policies are generated using the feedback coming from the model which executes the control actions [7], [8], [9]. The excepted reward of actions is correlated with the main KPIs, and learning strives to generate such control policies which maximize the expected reward on the long run. So far, two variants of this model can be distinguished: (1) *Order-oriented decomposition*, when the subjects (and targets) of learning are different decisions related to orders [8], and (2) *resource-oriented decomposition*, when resources are represented by autonomous agents and the control policies of these agents are learned individually [9]. A most recent work applies so-called deep Q-learning in order to manage the priority-based dispatching of orders in a complex wafer fab, complying also with strict time constraints [10]. For approximating the optimal action-value Q function, a deep convolutional neural network is used which circumvents the need of handcrafting the features of a large and unstructured state space [11]. This approach proved to be applicable in the scheduling of chemical production processes as well [12], and also for harmonizing the decisions of a combination of order and resource-oriented agents in a traditional manufacturing setting [13]. As broad-sweeping simulation experiments have shown, the results are encouraging, at least in relatively small-scaled problem instances.

At the same time, it is known that the combination of neural networks and reinforcement learning can be unstable, the training may diverge and, in general it is hard to provide performance guarantees for deep RL type methods [14]. Furthermore, as such approaches are getting momentum also in production control, they expose some dilemma: How can one guarantee that the control rules learned conforms with the intention of the system's designer? How can the black-box model be matched to reality? In the sensitive domain of stem cell production, we took a more stable and transparent approach by introducing a *policy gradient method* in combination with simulation to optimize the controller of an automated stem cell production platform.

Automated stem cell culture

Stem cells are important candidates for the medicine of the future, as they are offering new approaches towards understanding diseases, developing personalized medical treatments and can ultimately act as novel therapeutic agents themselves [15]. This generates a need for high quality stem cell material in sufficient quantities to meet the demands of research and clinics.

The cultivation and propagation of stem cells, however, is a complex and labour-intensive process. In contrast to conventional biopharmaceutical production, where one, well-defined biomolecule is generated by a well-characterized cell line, in stem cell production the product is the living cell itself. What is more, the raw cell material is derived from donors or the patients themselves, which results in high batch-to-batch variation with strong influence on the process performance [16].

This bears certain challenges towards the production, as the cultivation of stem cells requires agile production processes that are able to adapt to variations in the cell culture and can control the process accordingly. As the requirements and circumstances stress the limits of automation, manual labour is still highly prevalent in stem cell production. The consequences are elevated risk of contamination and increased variability between batches, which in turn puts the reproducibility of experiments or treatments based on the cells at risk.

Stem cell production facilities

In recent years, however, there have been clear signs of change in the industry. With the development of novel production technologies in the context of Industry 4.0, automation solutions are now increasingly developed and adopted [17]. The introduction of fully automated, robot-assisted systems that remove all direct interaction between the user and the product from the process reduces the risk of human failure and enables reproducible processes [18]. In addition to improvement in quality and quantity of cells produced, automation enables a more comprehensive monitoring of processes and the generation of an extensive data record, which helps manufacturers to align with the strict regulatory requirements.

Examples for robot-assisted systems that allow fully automated processing of stem cells include the following:

- The *AUTOSTEM* pipeline is designed for end-to-end automated production of therapeutic stromal cells in the multi-litre scale. The focus of the facility lies in generating cells at scale in alignment with Good Manufacturing Practice (GMP) and other regulatory requirements. In a follow-up project to the AUTOSTEM project, the facility is upgraded to allow production of multiple batches in parallel and connects the system with other automated modules for batch quality control and purification of extracellular vesicles [19].
- The *StemCellFactory* enables fully automated generation and expansion of clonal induced pluripotent stem cells (iPSC) from blood cells in a standardized and parallelized manner. In addition, it is possible to edit the genome of iPSC clones, which opens up a wide range of further possibilities towards studying disease and drug mechanisms [20], [21].
- The *StemCellDiscovery* represents a testbed for laboratory automation. Within the automated laboratory, new processes, devices, software and algorithms can be developed, tested and rolled out to stem cell production and beyond. The investigations described in the paper refer to this system (**Figure 1**).

Figure 1: The *StemCellDiscovery* testbed.

- The *iCellFactory* is an autonomous, easy-to-use testbed for cultivating induced pluripotent stem cells. It is highly reconfigurable and flexible. The iCellFactory can thus be used to develop and test novel processes, devices and software [22], [23].

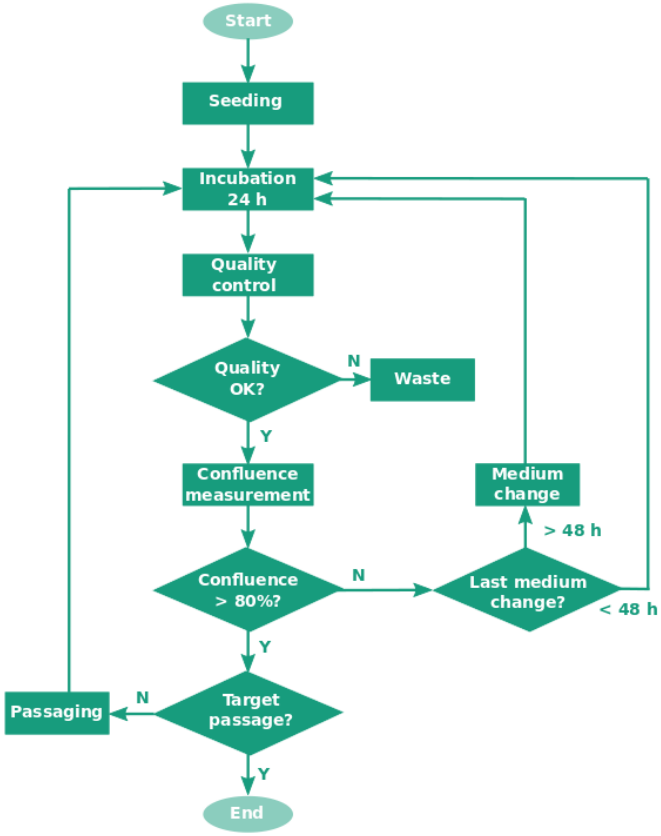

Figure 2: The workflow of stem cell production.

While the advances in production technology have shown to cope with the process variability and the resulting need for flexibility towards the process, the non-deterministic behaviour of the cells and thus the process still poses certain challenges for efficient, parallelized production of the cells in high throughput. The workflow of stem cell production is shown in **Figure 2.** This includes regular measurements, typically microscopic imaging to determine the cell density of the cell culture (percentage of the covered surface relative to total plate surface), which in the following will be referred to confluence [%].

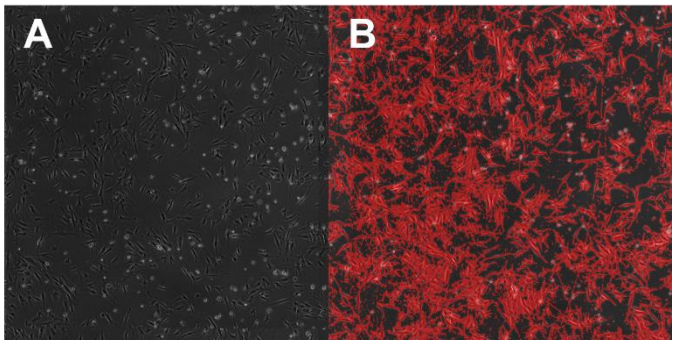

Figure 3: Microscopic image (A) and the result generated by the image processing algorithm (B).

Figure 3 illustrates a human Adipose-Derived Stem Cell (hADSC) culture at approximately 50 % normalised confluence. Image A is the microscopic image, B is the image with the overlay of the image processing algorithm detecting the cell culture confluence.

Depending on the measurement result, different process branches are taken in order to either only renew the nutrient media in which the cells are cultivated or harvest the cells for further processing if the cell density threshold is reached.

As the process is adaptive to the individual cell culture growth and behaviour, to date the process decision cannot be predicted prior to the measurement. However, this makes it difficult to generate and optimize job lists in advance, allowing only reactive scheduling approaches. In [24], a priority-based approach was presented that allows to take into account the specific constraints of certain cell cultures or processes. In efficient, high-throughput production a key issue is, however, the identification of optimal parameter sets which form the basis of scheduling and control decisions.

As the experimental efforts to generate sufficient data to optimize the process parameters are immense, there is also an opportunity of running validated simulation experiments. Specifically, one can simulate the production through a model that includes a representation of the product, process and the system itself. In the following, a simulation-based, intelligent optimization for bio-inspired control of the stem cell manufacturing process is presented.

Growth modelling

The crux of developing a model that is able to reproduce the variations in stem cell cultivation processes is capturing the cell growth behaviour. Cells in culture typically show a distinct growth behaviour that is divided into *three phases* [25]: (1) In the initial lag phase, freshly seeded cells are adjusting to their environment and therefore growing slowly. (2) In the exponential growth phase, cells are taking up their growth rate and multiply exponentially through cell division. (3) When one of the requirements for growth, such as nutrients or space, are depleting, the cell growth declines and cells enter the stationary phase.

This limited growth of the cells in culture reflects in growth curves with a sigmoidal shape, which is highly typical for biological populations. For this reason, there are many examples for mathematical descriptions of this growth behaviour. One of them is the Gompertz function, which was published in 1825 for the first time but since then has been used broadly to model the growth of cells [26]. The function has been recently revisited, and a unified mathematic equation with interpretable and comparable parameters has been derived [27]:

$$W_{(t)} = A\left(\frac{A}{W_0}\right)^{\exp(-e\cdot k_G\cdot t)} \quad (1)$$

The cell population $W_{(t)}$ is thereby a function of time and dependent from the upper limit for growth A, the initial cell density W_0 at time point $t = 0$ and the Gompertz growth constant k_G.

A different approach to modelling the cell growth is the Bertalanffy equation, which represents also a limited growth and according to [28], also allows to "accommodate crude 'metabolic types' based upon physiological reasoning". A unified version is given in [29]:

$$W_{(t)} = A\left(1 + \left(\left(\frac{W_0}{A}\right)^{\frac{1}{3}} - 1\right)\exp(-k_B\cdot t)\right)^3 \quad (2)$$

with the Bertalanffy growth constant k_B.

Since during the production process regular medium exchanges have to be performed, the main limiting factor is in the present case the availability of space. For the upper asymptote one can thus assume to be A = 100%, which is equivalent to a culture plate fully covered with cells (i.e., fully confluent). The parameters for W_0 and k, however, depend from culture to culture and thus introduce the variations in the growth behaviour. In order to generate a parameter set that reflects typical cultivations of stem cells, a dataset consisting of 72 growth curves was generated using stromal stem cells isolated from the adipose tissue of a 23-year-old female donor.

The cells were seeded at cell densities of 2.500 cells/cm² and subsequently cultivated in twelve sets of 6-well-plates for nine to thirteen days. Cells were grown in 3 mL per well Dulbecco's Modified Eagle Medium (DMEM), containing 10% foetal bovine serum (FBS) and 1% penicillin/streptomycin. The cells were cultivated in the incubator at 37°C in ambient air enriched with 5% CO_2. The cell culture was assessed regularly through microscopic imaging and the confluence derived from the images based on an image processing algorithm according to [30]. The numbers then were normalized so that the highest confluence measured represented 100%. To each data set, both the Gompertz and the Bertalanffy functions were fitted. Thereby A was fixed to 100% and W_0 was assumed to be the first confluence value measured in the time series.

Table 1: Parameter sets for simulation based on historical cell culture data (normalized for A = 100%).

	Low variability	Higher variability
W_0	14 % (12 – 16.0%)	20 % (10 – 30%)
k_{Gompertz} (k_G)	0.12 (0.09 – 0.15)	0.14 (0.08 – 0.2)
$k_{\text{Bertalanffy}}$ (k_B)	0.265 (0.2 – 0.33)	0.275 (0.15 – 0.4)
A	1 (100%)	1 (100%)
Threshold C_{Split}	0.8 (80%)	0.8 (80%)

As a result, average values for W_0 of 14% ± 2%, Gompertz coefficient k_G of 0.12 ± 0.03 and a Bertalanffy coefficient of k_B 0.265 ± 0.065 were obtained. Using the Gompertz or the Bertalanffy equation and the values for W_0 and k randomly generated within the given range, a large set of culture plates can be simulated that show typical growth behaviour, in this particular case, of stromal stem cells. As these data sets were acquired using data only from one donor and with the same media batch, one would expect an even higher variability between the cultures. Therefore, a second parameter set was also assumed that allows for higher variability. Both parameter sets are given in **Table 1**.

Simulation of the stem cell production

Getting good training data is one of the most difficult problems in machine learning (ML). This is especially true in biological applications, where experimenting on live cell cultures is not only expensive and unpractical, but it can also raise ethical issues. Simulation can be an inexpensive way to generate large volumes of training data, therefore, simulation-assisted ML has become more and more widespread in recent ML applications [31].

The intention was to build a simulation of the stem cell production factory in order to support learning the optimal control of the system. The main purpose was to generate large volume of data, therefore, the simulation was designed (1) to run fast, (2) to be easily configurable for different control policies, and (3) to provide aggregate performance indicators (such as output, waiting times or waste production).

Besides programmatically parameterizing and running the simulation, it is also important to validate the behaviour and the results of the simulation system. For this purpose, a graphical user interface has been designed that enables the control, monitoring and evaluation of the experiments with various charts, plots, animations and process logs (**Figure 4**).

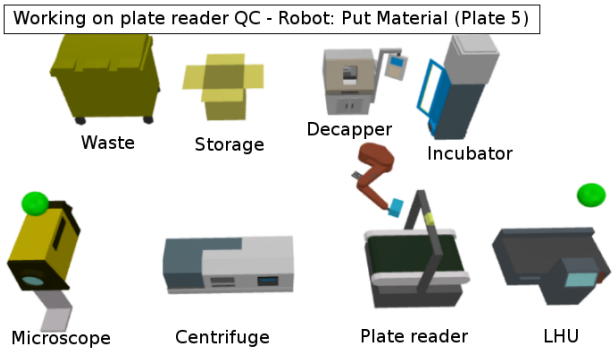

Figure 4: Simulation of the stem cell production.

Based on these requirements, the implementation of the simulation was made in [32], which provides a rich simulation toolkit, various visualization possibilities and is highly flexible and extendable. AnyLogic provides three different modelling paradigms: system dynamics, discrete event and agent-based, as well as it also supports multi-method modelling. The *agent-based paradigm* [33] was used for modelling the system components (such as plates and production equipment), while events triggered changes in the system (such as terminating an operation).

The equipment of the factory and the production processes have been built into the simulation model. Altogether five main processes were implemented in the simulation system: (1) the initial seeding, (2) the quality check, (3) the confluence measurement, (4) the medium exchange and (5) the distribution of the cell culture from one plate to multiple plates. Each process is defined by a flowchart and contains several consecutive steps. Some processes are even branch based on the system state, e.g., the length of the waiting queues. Each production step requires one of the processing equipment such as liquid handling unit (LHU), decapper, plate reader, centrifuge, microscope and incubator, while transportation steps use the robot arm.

Besides the fixed processes, the simulation can be customized with various parameters that can be set either in an external configuration file or by programmatically. These include the number, volume and growing rate of the cell cultures, the capacity of the factory, the duration of the processing steps, the various uncertainties in the system, as well as the inaccuracy of the measuring process.

In order to facilitate the learning of the optimal control, the decision logic can be configured in details, too. The key control factors include (1) the rules when and how the cell cultures should be distributed into multiple plates, (2) the threshold for initiation of medium exchanges, (3) the order of processing the jobs based on multi-part priorities, and (4) the maximum allowed waiting times outside of the incubator. These four factors imply the control

policy of the reinforcement learning algorithm, i.e., the policy specifies when the passaging and medium change processes should be executed, furthermore, if several plates are waiting for a resource, which one should be processed first (see also Section "Controller parametrization" for more details).

The state representation describes the status of the agents in the system, i.e., the plates and the resources. The state of the plates consists of (1) the confluence of each well of the plate, (2) whether the contents of the plate were already harvested, wasted or still in production, (3) number of passages that were performed on the plate, (4) time of the last medium change, (5) which of the five main processes, which step and for how long is being performed on the plate, (5) total time spent inside and outside of the incubator, as well as the total time spent waiting for resources since the beginning of the simulation. The state of the resources describes (1) whether it is idle or active, and if active, on which process and for how long it is working, (2) the waiting queue of the resource, and (3) the total working time and idle time since the beginning of the simulation.

Biologically inspired control of stem cell production

Biologically inspired computational approaches, such as artificial neutral networks, evolutionary algorithms, swarm intelligence and reinforcement learning (RL), are widely used in various applications, as these alternative methods are typically more flexible than traditional ones, and can successfully solve a wide range of optimization problems heavily burdened by uncertainties. We will be especially interested in RL methods [34], as they are well suited for resource allocation problems [35], [36], including scheduling and transportation, and can even efficiently handle time-varying environments [37].

In this section, after providing a brief overview of Markov decision processes (MDPs) and RL, we will describe how the controller of the automated stem cell production platform is parametrized, and how it is optimized by a policy gradient type RL method (based on the Kiefer-Wolfowitz stochastic approximation algorithm) via sequential interactions with the simulation environment.

Reinforcement learning

Reinforcement learning (RL) is one of the main branches of machine learning. It aims at optimizing the expected long-term (typically discounted or average) rewards an agent (controller, decision-maker) can achieve by interacting with an uncertain and dynamic environment [34]. Markov decision processes (MDPs) constitute the main underlying mathematical framework of RL [38]. MDPs and RL have a wide range of applications, from robot control and strategic asset pricing to communication networks and sequential clinical trials.

A *Markov Decision Process* (MDP) is a stochastic system defined by a 5-tuple $(\mathbb{X}, \mathbb{A}, U, p, r)$, where the components are as follows:

(1) $\mathbb{X}$ is the state space (measurable space);
(2) $\mathbb{A}$ is the action space (measurable space);
(3) $U: \mathbb{X} \to \mathcal{P}(\mathbb{A})$, where $\mathcal{P}(\mathbb{A})$ denotes the power set of $\mathbb{A}$, is the action constraint function (nonempty for all x ∈ $\mathbb{X}$);
(4) $p: \mathbb{X} \times \mathbb{A} \to \Delta(\mathbb{X})$, where $\Delta(\mathbb{X})$ is the space of all probability distributions over $\mathbb{X}$, is the transition probability function;
(5) $r: \mathbb{X} \times \mathbb{A} \to \Delta(\mathbb{R})$, where $\mathbb{R}$ denotes the field of real numbers, is the (possibly randomized) immediate reward function.

An MDP can be interpreted as follows. Consider an *agent* (decision maker) acting in a dynamic and uncertain *environment*. The agent receives information about the state of environment, $x \in \mathbb{X}$, and the available actions, $U(x)$, based on which it choses and action, $u \in U(x)$. After the decision was made, the state of the system changes according to the probability distribution, $p(x, u)$, and the agent receives a (real-valued) immediate cost or reward, $r(x, u)$.

Stochastic shortest path (SSP) problems are (undiscounted) MDPs in which there is an absorbing terminal state, $\tau \in \mathbb{X}$, such that for all $u \in U(\tau)$, we have $p(\tau, u) = \delta_\tau$, i.e., the point mass probability measure concentrated at τ, and $r(\tau, u) = 0$, with probability one. That is, once τ is reached, the agent stays there forever without incurring any more rewards, which can be interpreted as the process terminates [39].

For the case of the automated stem cell production platform, the state representation is discussed at the end of Section "Simulation of the stem cell production". Of course, the terminal state represents the end of the production process.

The behaviour (decision strategy) of the agent is described by its control *policy*, which is a (possibly randomized) mapping from states to (control) actions, $\pi: \mathbb{X} \to \Delta(\mathbb{A})$. A policy is called *proper*, if from all starting states, the expected number of steps needed to reach the terminal state τ is finite. A key concept in MDPs is the *value function*, which shows how much total rewards the agent can expect starting from a given state and following the given policy thereafter. Formally, the *value function* of proper policy π is

$$V^\pi(x) \stackrel{\text{def}}{=} \mathbb{E}\left[\sum_{t=0}^{\infty} r(X_t, U_t) \,\middle|\, X_0 = x \right],$$

where $X_{t+1} \sim p(X_t, U_t)$, $U_t \sim \pi(X_t)$, with "~" is the abbreviation of "has distribution". Note that, for simplicity, we will assume that the rewards are bounded, hence, as the policy is assumed to be *proper*, the value function is well-defined for all possible state x. Also note that finite horizon MDPs are special cases of SSP problems.

The decision logic applied in the automated stem cell production platform is discussed in Section "Simulation of the stem cell production", while the control policy and its parametrization is further explained in Section "Controller parametrization".

One of the fundamental concept of studying MDPs is the Bellman optimality operator. It has the form $T: \mathfrak{B}(\mathbb{X}) \to \mathfrak{B}(\mathbb{X})$, where $\mathfrak{B}(\mathbb{X})$ is the set of all bounded functions over set $\mathbb{X}$, and it is defined as

$$(TV)(x) \stackrel{\text{def}}{=} \max_{u \in U(x)} \mathbb{E}[\, r(x, u) + V(y) \,],$$

where $y \sim p(x, a)$, and assuming that the maximum is always well defined, for example, $U(x)$ is finite for all state x.

In case the SSP problem is finite (namely, $|\mathbb{X}| + |\mathbb{A}| < \infty$), and all policies are proper, the Bellman operator is a contraction in the weighted maximum norm. The optimal value function, V^*, is then the unique solution of the Bellman optimality equation,

$$TV^* = V^*,$$

and although there could be many optimal control policies, they all share the same unique optimal values function, V^*.

The three classical ways of solving finite SSP problems are as follows: (1) iteratively approximating the optimal value function by a sequence of value functions (e.g., value iteration); (2) directly searching in the space of policies (e.g., policy iteration); and (3) reformulating the SSP problem as a (static and deterministic) linear programming optimization problem [39], [34].

A main drawback of the aforementioned classical solutions are that they presuppose the complete knowledge of the underlying system (for example, the transition probabilities) which is typically not available in practice; moreover, they are only practical for MDPs with small state spaces. On the other hand, RL methods address the problem when a model of the system is unavailable and the agent should simultaneously explore the environment via interactions and improve its decision strategy (control policy). Many RL methods are also able to efficiently deal with infinite MDPs.

It is often the case that though a precise mathematical model of the system based on which one could formulate an optimal policy is not available, however, the system can be simulated and we can train a controller based on a large number of simulations.

In case of *Cyber-Physical Production Systems* (CPPS) [40], creating a digital twin is often feasible, which then can be used to optimize a controller based on RL methods. This approach was used in the case of the addressed automated stem cell platform: a simulation environment was created (as described in Section "Simulation of the stem cell production"), and then an RL based system was connected to the simulation. The controller was then iteratively refined via sequential interactions with the simulated environment.

Figure 5 illustrates the interaction of the controller with the simulated environment. The state at time t is denoted by x_t, the control action is u_t, while the incurred cost or reward is c_t. The dynamics of the environment is also driven by a random noise, n_t, representing the uncertainties affecting the system.

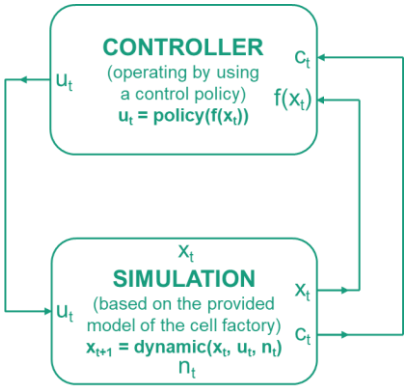

Figure 5: The interaction between the controller and the simulated (dynamic and uncertain) environment based on state and reward feedbacks.

Controller parametrization

The original controller of the automated stem cell platform is based on a priority rule. Specifically, to each plate a priority is assigned, which can also change over time; and the higher is the priority of a plate, the earlier it is served by an available resource. The overall priority of a plate is calculated by a weighted sum of the following five features:

1. Process priority
2. Falcon tube priority
3. Plate waiting time
4. Plate confluence weight
5. Time spent outside of the incubator

Only the features "plate waiting time", "time spent outside of the incubator", and "plate confluence" change during the simulation, the values of the other two features are fixed in advance. The behaviour of the stem cell platform is also influenced by the following two important thresholds:

6. Confluence threshold for splitting
7. Minimum incubation time between measurements

The "confluence threshold" is the value after which the cell culture is divided and split between two new falcon tubes. The "incubation time" is a threshold describing the minimum time that must be spent between two confluence measurements.

The weights of the five features above (1-5) and the values of the two thresholds (6-7) will be referred to as the parameters of the controller, as these seven numbers determine the behaviour of the system. In the original system, these seven parameters were set based on the domain specific knowledge of experts. On the other hand, the main aim was to exploit that a simulation model is available and apply an RL method to optimize these parameters by using feedback from sequential simulation scenarios. Henceforth, the vector of these parameters will be denoted by $w \in \mathbb{R}^d$, where the dimension, d, depends on how many of these parameters are actually optimized (see, for example, Table 2 of Section "Biologically inspired control of stem cell production").

Controller optimization

The aim of the optimization process is to maximize the expected throughput – namely, the mean cell yield over a specified horizon – of the stem cell platform. In order to optimize the parameters of the controller via interactions with the simulation model, a policy gradient type RL algorithm is applied. Policy gradient (PG) is a class of RL methods that are variants of the stochastic gradient algorithm for MDPs. One of their advantages is that they can deal with infinite, continuous state and action spaces. There are various PG methods available, e.g., the REINFORCE algorithm is a popular choice [34]. However, REINFORCE presupposes a differentiable control policy parametrization with known derivatives, which was not available in our case. Hence, we have applied the Kiefer-Wolfowitz (KW) type stochastic approximation method instead [41]. **Figure 6** overviews the KW method for MDPs.

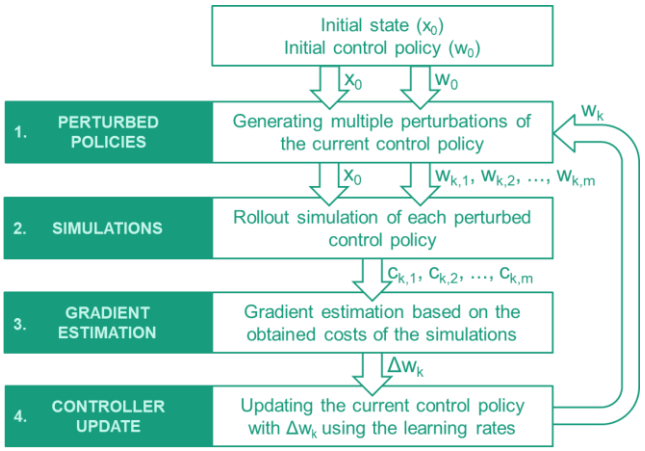

Figure 6: Policy gradient based on the Kiefer-Wolfowitz stochastic approximation method for MDPs: Iteratively approximating an optimal controller via estimating the gradient of the value function based on simulations with policies having perturbed parameters.

Policy gradient methods iteratively refine the parameters of the controller [34], denoted by w_k for iteration k in **Figure 6**. In case of the KW method, in each iteration it makes simulations with slightly

perturbed controller parameters. The simulations result in value function estimates for those perturbed parameters, denoted by $c_{k,m}$ for iteration k and simulation m. These values then can be used to estimate the gradient of the value function, with respect to the control parameters, at the current controller configuration. Having (a noisy estimate of) the gradient vector at hand, the method updates the controller configuration by making a "small" step towards the direction of the gradient.

More precisely, let $R(w, \varepsilon)$ denote the (random) total rewards gathered during a trial (simulation) by running a control policy parametrized by $w \in \mathbb{R}^d$ from a fixed initial state, x_0, until the terminal state, τ, is reached, where ε is a random element encoding all the uncertainties generated during the simulation of the SSP problem; i.e., after ε is chosen, $R(\cdot, \varepsilon)$ is deterministic.

Note that, using the previous notations, $V^{\pi}(x_0) = \mathbb{E}\,[\,R(w, \varepsilon)\,]$, where π is the control policy parametrized by $w \in \mathbb{R}^d$. We will also use the simplified notation $V(w) \stackrel{\text{def}}{=} \mathbb{E}\,[\,R(w, \varepsilon)\,]$.

In case of the stem cell production platform, $R(w, \varepsilon)$ is the total cell yield over a given horizon: the sum of confluence in all plates at the end of the simulation. The initial state describes the starting configuration of the platform at the start of the optimization.

A crucial step of PG methods is that they need to estimate the gradient of the value function at a given controller parametrization $w \in \mathbb{R}^d$. As the gradient is a vector containing the partial derivatives w.r.t. all possible coordinate directions, we need to estimate the partial derivatives. The KW method estimates the partial derivative of the value function w.r.t. the i$^{\text{th}}$ parameter by

$$\left(\frac{\partial V}{\partial w_i}\right)(w) \;\approx\; D(w, i, \Delta, \varepsilon^{\circ}, \varepsilon_i^{+}) \;\stackrel{\text{def}}{=}\; \frac{R(w + e_i\Delta, \varepsilon_i^{+}) - R(w, \varepsilon^{\circ})}{\Delta},$$

where e_i is the standard unit vector in the i$^{\text{th}}$ coordinate direction, $\Delta > 0$ determines the size of the finite difference interval used to estimate the partial derivative, and ε° and ε_i^{+} are two independent random elements encoding the uncertainties during two different simulations of the MDP. Note that we use the one-sided estimate for the partial derivatives, in order to decrease the number of simulation runs needed to estimate the gradient. KW estimates the gradient by estimating each partial derivative one by one, that is

$$(\nabla V)(w) \;\stackrel{\text{def}}{=}\; \begin{bmatrix} \left(\frac{\partial V}{\partial w_1}\right)(w) \\ \vdots \\ \left(\frac{\partial V}{\partial w_d}\right)(w) \end{bmatrix} \;\approx\; G(w, \Delta, \varepsilon) \;\stackrel{\text{def}}{=}\; \begin{bmatrix} D(w, 1, \Delta, \varepsilon^{\circ}, \varepsilon_1^{+}) \\ \vdots \\ D(w, d, \Delta, \varepsilon^{\circ}, \varepsilon_d^{+}) \end{bmatrix},$$

where $\varepsilon = (\varepsilon^{\circ}, \varepsilon_1^{+}, \cdots, \varepsilon_d^{+})$ is a vector of (i.i.d.) random elements encoding the uncertainties of the $d + 1$ simulations needed to estimate the gradient vector at a given point $w \in \mathbb{R}^d$.

Using the notations above, the KW algorithm proceeds as

$$w_{k+1} \;=\; w_k \;+\; \gamma_k\, G(w_k, \Delta_k, \varepsilon_k),$$

where w_k is the parameter vector of the control policy, γ_k is the learning rate or step-size, Δ_k is the length of the finite difference interval used to estimate the partial derivatives, and ε_k is the uncertainty vector of the simulations at iteration k.

The learning rates, $\{\gamma_k\}$, and the lengths of the finite difference intervals for the gradient estimation, $\{\Delta_k\}$, must satisfy

$$\sum_{k=1}^{\infty} \gamma_k = \infty, \qquad \lim_{k\to\infty} \Delta_k = 0, \qquad \sum_{k=1}^{\infty} \frac{\gamma_k^2}{\Delta_k^2} < \infty,$$

with typical choices $\gamma_k = a \cdot k^{-1}$ and $\Delta_k = b \cdot k^{-1/4}$, for $a, b > 0$. Then, under mild regularity conditions [41]

$$(\nabla V)(w_k) \xrightarrow{a.s.} 0, \qquad \text{as} \qquad k \to \infty,$$

that is, the controller parameters, $\{w_k\}$, converge with probability one to a stationary point of the gradient of the value function.

Note that, in general, the convergence of such gradient methods is only guaranteed to a local optimum. On the other hand, by restarting the optimization from random initial parameters, the probability of reaching the global optimum can be increased.

Results of the reinforcement leaning-based controller in a simulated environment

Altogether fifteen numerical experiments were initiated and carried out for three system sizes (with the capacity of 50, 100 and 200 plates, respectively), in order to compare four possible types of optimization scenarios (see below) as well as the original controller parameters (based only on expert knowledge, without having any additional optimization).

The simulation scenarios always started with 18 initial falcon tubes, each having possibly different cell growth properties, and lasted for 20 (virtual) days. The largest system (with capacity of 200) could accommodate all plates (created by splitting) in every simulation, thus its results would be the same for larger systems.

The following approaches were compared:

1. Default parameters set by experts (not optimized)
2. Plate priority optimization (only the weights of the priority rule were optimized)
3. Confluence threshold optimization (only one threshold was optimized, the others were set to default values).
4. Incubation time optimization (only one threshold was optimized, the default values were used for the others)
5. Full optimization of all seven control parameters

Regarding the experiments involving RL-based optimization (i.e., approaches 2-5), 100 iterations were performed in each case, as it turned out that such a low number of iterations were enough to ensure close-to-optimal solutions. As the simulations were noisy, the parameters could only reach a (small) neighbourhood of the optimal parameters after finite number of iterations.

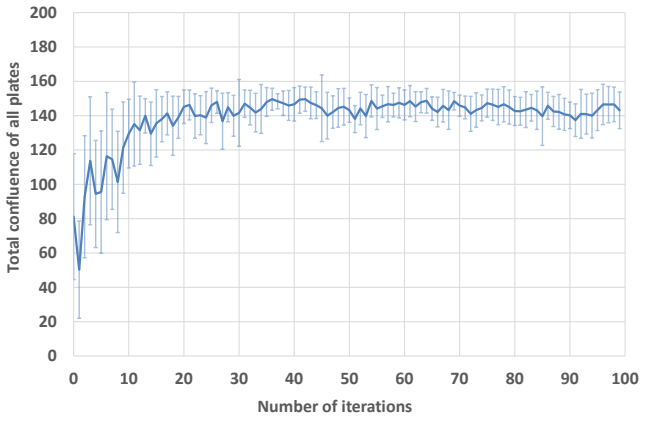

Figure 7: Average and standard deviation of ten learning curves in case all parameters were optimized, there were maximum 200 plates allowed in the system and each optimization started from a random initial configuration of controller parameters.

Table 2: Control parameters with performance data for each scenario.

Experiment	System		Plate priority weights					Decision parameters		Objective	
	Nr of tubes	Max plate in system	Process weight	Tube weight	Waiting time weight	Confluence weight	Time spent outside	Confluence threshold for splitting	Measurement interval in days	Average output (sum of confluence in all plates)	Deviation of outputs (sum of confluence in all plates)
Not-optimized (default)	18	50	1	0	1	0	0	0.8	1	72.347	2.154
Not-optimized (default)	18	100	1	0	1	0	0	0.8	1	105.768	4.097
Not-optimized (default)	18	200	1	0	1	0	0	0.8	1	117.662	4.486
Priority optimized	18	50	0.17	1.54	2.77	0.3	0.87	0.8	1	72.982	2.200
Priority optimized	18	100	2.07	2.02	2.78	2.58	0.37	0.8	1	107.798	2.880
Priority optimized	18	200	2.5	2.52	2.9	1.91	1.36	0.8	1	116.842	4.928
Confluence threshold optimized	18	50	1	0	1	0	0	0.73	1	76.756	1.850
Confluence threshold optimized	18	100	1	0	1	0	0	0.69	1	121.102	3.079
Confluence threshold optimized	18	200	1	0	1	0	0	0.68	1	143.976	5.986
Incubation time optimized	18	50	1	0	1	0	0	0.8	0.22	71.206	4.145
Incubation time optimized	18	100	1	0	1	0	0	0.8	1.24	105.512	2.699
Incubation time optimized	18	200	1	0	1	0	0	0.8	0.83	121.715	4.587
Fully optimized	18	50	0.31	2.82	2	2.42	0.2	0.65	1.81	81.136	1.942
Fully optimized	18	100	1.26	2.53	0.07	2.66	0.09	0.68	0.69	122.023	3.132
Fully optimized	18	200	2.09	0.08	0.16	0.44	2.53	0.68	1.64	152.613	5.029

Figure 7 shows the average and standard deviation of ten different learning curves demonstrating that the controller parameters typically stabilized after as few as about 50 iterations of the optimisation procedure. The plotted standard deviations (vertical bars) show that even though random starting configurations were applied, all experiments converged to controllers with very similar performances. This is indicative of the phenomenon that the optimization is robust with respect to the choice of the starting configuration (and less likely to stuck in local minima).

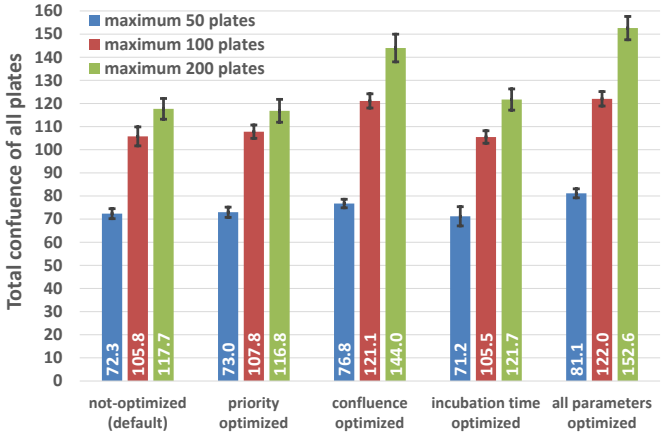

Figure 8: Comparing the averages and standard deviations of total confluences for optimizing various groups of parameters.

Figure 8 and **Table 2** display the optimized values of the control parameters for each experiment as well as the corresponding average total confluences with their standard deviations. By "total confluence" we mean the sum of all confluence parameters in all plates at end of the simulation. The presented standard deviations were calculated based on the last 50 iterations of the process.

The results show that optimizing the two thresholds have a larger influence on the throughput of the system than only optimizing the weights of the priority rule. Moreover, the performance increase is larger for systems with larger capacity constraints. It can also be observed that the confluence threshold, that determines at which cell density the culture is harvested, is the most significant control parameter: it produced the largest change in the throughput, even if the other parameters were fixed to their default values. Based on Table 2, one can conclude that the best results were achieved when this parameter was set to about 68 – 70 %. Overall, the best performance was attained when all seven control parameters were optimized and the capacity of the system was the largest, i.e., 200 plates. In this case, the average throughput (compared to the default parameters set by experts) was increased by about 30%, which clearly demonstrates the viability and efficiency of the presented RL approach. Table 2 shows the standard deviations of the parameters, as well. They indicate that the control parameters stabilized well around the obtained controller configurations.

As a simulation model is only an approximation of reality, it is crucial to study how *robust* the obtained controller is, in case it is applied for slightly different problems than it was optimized on. Therefore, a *sensitivity analysis* was initiated during which we have studied how the optimized controller performs, when we assume particular growth model parameters during the learning process, but the actual system behaves according to another model.

For these experiments, we worked with the *Bertalanffy* cell growth model. During the optimization of the controller, the simulation environment applied a *nominal* growth model with parameters $k_B = 0.275$ and $W_0 = 0.2$. However, we tested the controller on systems that behaved differently; namely, the cells grew according to a model with potentially different k_B and W_0 parameters.

Figures 9 and **10** compare the performances of our optimized controller to that of the default (non-optimized) one, for various grown model parameters, in case the optimization was only done according to the aforementioned nominal model. Using a baseline solution (i.e., the default controller) is important for comparison, as changing the cell growth parameters has a significant effect on the achievable cell yield, thus the absolute performance values are less meaningful than the relative ones compared to a baseline.

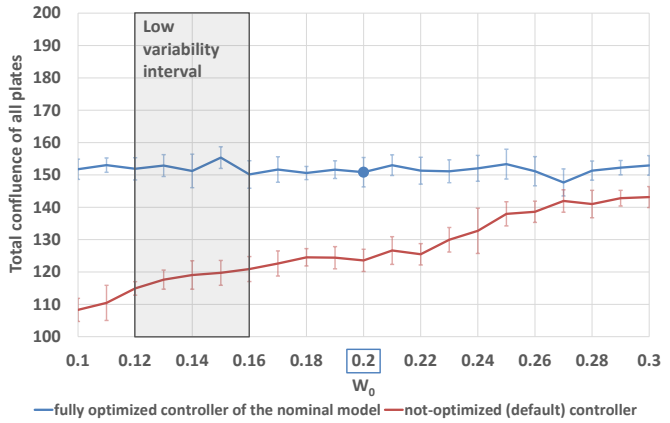

Figure 9: Sensitivity analysis of the controller for the W_0 initial population parameter, the optimization assumed $W_0 = 0.2$ (and $k_B = 0.275$). The averages and standard deviations are shown.

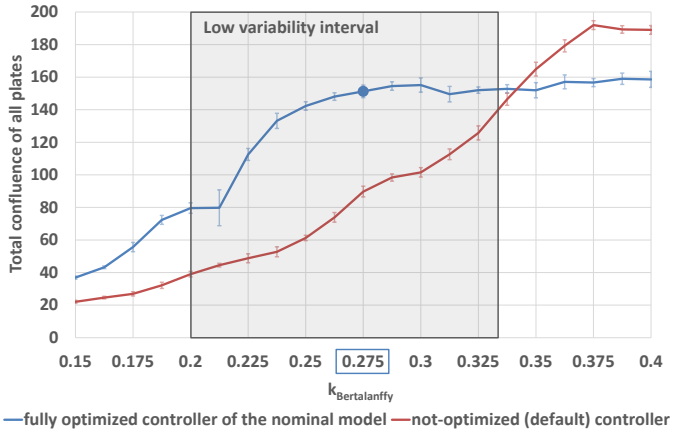

Figure 10: Sensitivity analysis of the controller for the k_B growth parameter. The optimization assumed $k_B = 0.275$ (and $W_0 = 0.2$). The averages and standard deviations are shown.

For each tested parameter value, the shown performance scores are averages of ten simulations. The vertical bars indicate the standard deviations of the ten performance realizations. The simulations started with 18 plates and the maximum allowed plates in the system was set to 200 in all cases.

The controller optimized on the nominal model outperformed the baseline for all allowed parameter values (cf. Table 1), in case of imprecise initial population values. For the case of imprecise growth parameter, the optimized one outperformed the baseline for the low variability interval, and was only worse for a part of the high variability interval. These results indicate that the suggested approach is robust against imprecisely estimated growth models.

Conclusion

For a time, Cyber-Physical Production Systems are considered to have faculties which can open new avenues for controlling highly complex manufacturing systems, even in dynamic and uncertain environments [40]. One of these novel options with a strong future application potential is biologicalisation, or the biological transformation in manufacturing [42]. According to the authors of the above paper, biologicalisation is "the use and integration of biological and bio-inspired principles, materials, functions, structures and resources for intelligent and sustainable manufacturing technologies and systems with the aim of achieving their full potential".

The research reported in the paper is an exemplar of biologicalisation in its own right, i.e. the use of bio-inspired algorithms for controlling a manufacturing system which produces biological material. In this setting, the automation of the production of stem cells faces a number of challenges. However, as the first results suggest, a well-fit growth model of cell cultures, combined with an agent-based simulation model of the all the main objects and resources in this micro-world of production can provide a reliable basis for a reinforcement learning-based control scheme. This novel approach, even though it is data-intensive, gives room for incorporating existing background knowledge of the application domain, and, at the same time, can enhance the performance of actual solutions using rule-based control.

The general conclusions arranged under four headings are:

- *Automated production of biological materials* such as of *stem cells* represents perhaps the highest level of biological transformation in manufacturing, where a *symbiotic co-existence* and of *co-evolution* of the technical, ICT and biological compounds are manifested.
- The in-depth *interconnection of technology and biology* holds new challenges, but also opportunities for *completely new possibilities* for solutions.
- *Bridge building* was demonstrated between discrete part manufacturing science and technology, on the one hand, and biological / medical sciences, on the other.
- *Biologically inspired algorithms* such as reinforcement learning can bring significant benefits for designing, planning and controlling production systems which are operating under inherent uncertainties.

Future research will focus on the implementation and extensive testing of the novel RL-based control scheme in an experimental automated stem cell production environment where the agent-based simulation is extended to a digital twin. We are convinced that the main principles and core solution techniques can be transferred also to the production of other bio-materials as well as to more traditional domains of manufacturing where the production processes are burdened even by great uncertainties.

Acknowledgements

The authors wish to thank the Fraunhofer Gesellschaft and Professor Reimund Neugebauer, President of the Fraunhofer Gesellschaft for the initiation and proactive support with this project. We also acknowledge the support of the Fraunhofer Team at the Think-Tank under the leadership of Dr.-Ing. Sophie Hippmann. Ms. Kerstin Funck (Fraunhofer), Dr. James Ryle (UCD) and Mr. Philip Meagher (UCD) provided excellent support with this international project.

The theoretical background of the biologically inspired control was elaborated within the GINOP-2.3.2-15-2016-00002 project. The research done by SZTAKI was also supported by the Ministry of Innovation and Technology NRDI Office within the framework of the Artificial Intelligence National Laboratory Program.

References

[1] Robinton, D.A., Daley, G.Q., (2012), The promise of induced pluripotent stem cells in research and therapy, *Nature*, 481: 295–305.

[2] Mount, N.M., Ward, S.J., Kefalas, P., Hyllner, J., (2015) Cell-based therapy technology classifications and translational challenges, *Philos Trans R Soc Lond B Biol Sci*, 370: 1680.

[3] Williams, D.J., Thomas, R.J., Hourd, P.C., Chandra, A., Ractliffe, E., Liu, Y., Rayment, E.A., Archer, J.R., (2012), Precision manufacturing foor clinical-quality regenerative medicines, *Philosophical Transactions of The* Royal *Society A*, 370: 370:3924-3949.

[4] Egri, P., Csáji, B.Cs., Kis, K.B., Monostori, L., Váncza, J., Ochs, J., Jung, S., König, N., Schmitt, R., Brecher, C., Pieske, S., Wein, S., (2020), Bio-inspired control of automated stem cell production, *Procedia CIRP*, 88: 600-605.

[5] Kádár, B., Lengyel, A., Monostori, L., Suginishi, Y., Pfeiffer, A., Nonaka, Y., (2010), Enhanced control of complex production structures by tight coupling of the digital and the physical worlds, *CIRP Annals – Manufacturing Technology*, 59(1): 437-440.

[6] Kuhnle, A., Lanza, G., (2019), Application of reinforcement learning in production planning and control of cyber physical production systems, In: *Machine Learning for Cyber Physical Systems*, Springer: 123-132.

[7] Lubosch, M., Kunath, M., Winkler, H., (2018), Industrial scheduling with Monte Carlo tree search and machine learning, *Procedia CIRP*, 72: 1283-1287.

[8] Stricker, N., Kuhnle, A., Sturm, R., Friess, S., (2018), Reinforcement learning for adaptive order dispatching in the semiconductor industry, *CIRP Annals – Manufacturing Technology*, 67(1), 511-514.

[9] Waschneck, B., Reichstaller, A., Belzner, L., Altenmüller, T., Bauernhansl, T., Knapp, A., Kyek, A., (2018), Optimization of global production scheduling with deep reinforcement learning, *Procedia CIRP*, 72: 1264-1269.

[10] Altenmüller, T., Stüker, T., Waschneck, B., Kuhnle, A., Lanza, G., (2020), Reinforcement learning for an intelligent and autonomous production control of complex job-shops under time constraints, *Production Engineering*, 14:319-328.

[11] Mnih, V., Kavukcuoglu, K., Silver, D. et al., (2015), Human-level control through deep reinforcement learning, *Nature*, 518:529-533.

[12] Hubbs, C.D., Li, C., Sahinidis, N.V., Grossmann, I.E., Wassick, J.M., (2020), A deep reinforcement learning approach for chemical production scheduling, *Computers & Chemical Engineering*, 141, 106982.

[13] Dittrich, M.A., Fohlmeister, S., (2020), Cooperative multi-agent system for production control using reinforcement learning, *CIRP Annals – Manufacturing Technology*, 69(1):389-392.

[14] Tsitsiklis, J. N., Van Roy, B. (1997). An analysis of temporal-difference learning with function approximation. IEEE Transactions on Automatic Control, 42(5), 674-690.

[15] Squillaro, T., Peluso, G., Galderisi, U., (2016), Clinical trials with mesenchymal stem cells: An update, *Cell transplantation*, 25(5): 829–848.

[16] Heathman, T.,R.J., Rafiq, Q.A., Chan, A.K.C., Coopman, K., Nienow, A.W., Kara, B., Hewitt, C.J. (2016): Characterization of human mesenchymal stem cells from multiple donors and the implications for large scale bioprocess development, *Biochemical Engineering*, *Journal* 108(15): 14–23.

[17] Brecher, C.; Malik, A.; Blanke, P.; Herfs, W., (2019), Dynamic integration of manual and automated biological process skills into MES, *24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Zaragoza, Spain, 10-13 September, 2019: 1069-1076.

[18] Kulik, M., Ochs, J., König, N., Schmitt, R., (2016), Automation in the context of stem cell production – where are we heading with Industry 4.0?, *Cell Gene Therapy Insights*, 2(4): 499-506.

[19] AUTOSTEM – stem cell manufacture, http://www.autostem2020.eu/, downloaded at July 22, 2019.

[20] https://www.stemcellfactory3.de/, downloaded at July 23, 2019.

[21] Marx, U., Schenk, F.; Behrens, J., Meyr, U., Wanek, P., Zhang, W., (2013), Automatic production of induced pluripotent stem cells, *Procedia CIRP*, 5: 2-6.

[22] Brecher, C., Wein, S., Xu, X., Storms, S., Herfs, W., (2019), Simulation framework for virtual robot programming in reconfigurable production systems, *Procedia CIRP*, 86: 98-103.

[23] Brecher, C., Pieske, S., Malik, A., Storms, S., (2019), Modelling of devices in an adaptive and dynamic environment, *Procedia CIRP*, 86: 210-215.

[24] Kulik, M., Ochs, J., Niels, K., McBeth, C., Sauer-Budge, A., Sharon, A., Schmitt, R., (2017), Parallelization in automated stem cell culture, *Procedia CIRP*, 65: 242-247.

[25] Butler, M., Animal cell culture and technology. Taylor & Francis, 2003.

[26] Gompertz, B., (1825), XXIV. On the nature of the function expressive of the law of human mortality, and on a new mode of determining the value of life contingencies. In a letter to Francis Baily, Esq. F. R. S. &c., *Phil. Trans. R. Soc*, 115: 513-583.

[27] Tjørve, K.M.C., Tjørve E., (2017), The use of Gompertz models in growth analyses, and new Gompertz-model approach: An addition to the Unified-Richards family, *PloS one*, 12(6): e0178691.

[28] Tsoularis, A., Wallace, J., (2002), Analysis of logistic growth models, *Mathematical Biosciences*, 179(1): 21-55.

[29] Tjørve, E., Tjørve K.M.C., (2010), A unified approach to the Richards-model family for use in growth analyses: Why we need only two model forms, Journal of theoretical biology, 267(3) :417–425.

[30] Schenk, F.W., Kulik, M., Schmitt, R., (2015), Metrology-based quality and process control in automated stem cell production, *tm - Technisches Messen*, 82(6): 309–316.

[31] Deist, T.M., Patti, A., Wang, Z., Krane, D., Sorenson, T., Craft, D., (2019), Simulation-assisted machine learning, *Bioinformatics*, 35(20): 4072–4080.

[32] The AnyLogic Company. AnyLogic Simulation Modeling Software Tool 2020, (https://www.anylogic.com/.), downloaded at March 20, 2020.

[33] Monostori, L., Váncza, J., Kumara, S.R., (2006), Agent-based systems for manufacturing, *CIRP Annals – Manufacturing Technology*, 55(2): 697-720.

[34] Sutton, R.S., Barto, A.G., Reinforcement learning: An introduction, 2nd edition, Cambridge, MA, The MIT Press; 2018.

[35] Csáji, B.Cs., Monostori, L., (2008), Adaptive stochastic resource control: A machine learning approach, *Journal of Artificial Intelligence Research (JAIR)*, 32: 453–486.

[36] Csáji, B.Cs., Adaptive resource control: Machine learning approaches to resource allocation in uncertain and changing environments. PhD. Thesis, Eötvös Loránd University, Budapest, Hungary, 2008.

[37] Csáji, B.Cs., Monostori, L., (2008), Value function based reinforcement learning in changing Markovian environments, *Journal of Machine Learning Research (JMLR)*, 9: 1679-1709.

[38] Bäuerle, N., Rieder, U., Markov Decision processes with applications to finance, Springer Science & Business Media, 2011.

[39] Bertsekas, D. P., Tsitsiklis, J. N.: Neuro-Dynamic Programming, Athena Scientific, 1996

[40] Monostori, L., Kádár, B., Bauernhansl, T., Kondoh, S., Kumara, S.R.T., Reinhart, G., Sauer, O., Schuh, G., Sihn, W., Ueda, K., (2016), Cyber-physical systems in manufacturing, *CIRP Annals – Manufacturing Technology*, 65(2): 621-641.

[41] Yin, G. G., Kushner, H. J. (2003), Stochastic approximation and recursive algorithms and applications, 2nd edition, Springer.

[42] Byrne G., Dimitrov, D., Monostori, L., Teti, R., van Houten, F., Wertheim, R., (2018), Biological transformation in manufacturing, *CIRP Journal of Manufacturing Science and Technology*, 21: 1-32.