Journal of Intelligent Manufacturing
https://doi.org/10.1007/s10845-023-02173-6

Developing an artificial neural network controller for accelerating the hot deformation of the titanium aluminide TNM-B1 using reinforcement learning and finite element simulations

J. A. Stendal[1] · M. Bambach[2,3]

Received: 9 May 2022 / Accepted: 21 June 2023

Abstract

This work presents a framework for interfacing a reinforcement learning algorithm with a finite element model in order to develop an artificial neural network controller. The goal of the controller is accelerating the hot compression process of the titanium aluminide TNM-B1. The reinforcement learning algorithm interacts with the finite element model by exploring different die velocities and receiving input measurements (the velocity, displacement and force of the die) while collecting rewards if a constant stress state in the workpiece is achieved. Synthetic stochastic material behavior was used to simulate the observed variations in deformation behavior of TNM-B1. The same reinforcement learning setup and reward function was able to adapt to two example finite element environments; the compression of a simple cylinder workpiece between flat dies and the compression of a more complex bone workpiece between flat dies. The performance of the controller for the bone compression environment was comparatively reduced and less consistent. In addition, training times and training instability were significantly increased. Furthermore, the results suggest that the framework can be used as a tool to find process optimizations or alternative process routes. This work demonstrates the concept and provides the groundwork and fundamentals for transferring the method to a physical setup.

Keywords Reinforcement learning · Finite element method · Process control · Deformation acceleration · Synthetic material behavior

Introduction

Reinforcement learning (RL) algorithms can learn to perform a task by interacting with a simulated environment. For instance, an RL algorithm can learn to master chess through self-play on a simulated board (Silver et al., 2018). In science and engineering, the finite element method (FEM) is commonly used as a tool to simulate and solve problems. By interacting with FE environments, RL algorithms can be used to identify optimizations, discover alternative process routes and to develop process controllers directly from the simulation. Industrial process control involves monitoring and adjusting key process variables in order to achieve a specific goal. In the metal forming industry, variations in the initial workpiece geometry, material behavior, temperature and lubrication conditions can result in varied mechanical properties and geometry, which in turn can affect the quality of the final product. Thus, controlling the parameters of metal deformation processes is central to obtaining desirable mechanical properties and consistent product quality at minimized costs (Kumar et al., 1992; Allam et al., 2014). Traditional control methods range from simple proportional-integral-derivative (PID) control that minimizes the difference between a measured variable and a target value, to complex model predictive control (MPC) that uses physics-based modeling to predict future process states and manipulates the variables accordingly (Camacho &

✉ J. A. Stendal
johan.stendal@sintef.no

M. Bambach
mbambach@ethz.ch

[1] Department of Materials Technology, SINTEF Manufacturing AS, Grøndalsvegen 2, Bygning 100, 2830 Raufoss, Norway

[2] Advanced Manufacturing Lab, ETH Zürich, Rämistrasse 101, 8092 Zurich, Switzerland

[3] Fraunhofer IGCV, Am Technologiezentrum 10, 86159 Augsburg, Germany

Published online: 20 September 2023

Alba, 2013). Developing an MPC controller requires knowledge of the underlying physics and dynamics governing the process, formulating a mathematical model of the process and finally deriving a control law that connects the measured input process variables to the output adjustments. Alternatively, RL can be used to develop the controller directly from a model or a simulation. An RL algorithm learns to map input variables to output adjustments by interacting with an environment through a process of trial and error, learning to perform the optimal actions according to a given measure of reward. This learning process can require many training iterations before a desirable solution is found. Consequently, RL algorithms are often trained in simulated environments rather than physical environments, as the learning process involves many costly and time consuming mistakes in a physical process. After achieving desirable performance in the simulated environment, the RL controller can be transferred to the real environment. Using FE simulations as the simulated environment, an RL algorithm can learn to control a deformation process according to a given reward system. Utilizing FE simulations to develop analytical controllers has previously been discussed in the literature (Zhang et al., 2016; Mikail et al., 2013). Furthermore, the relationship between RL and MPC have previously been investigated (Görges, 2017; Ernst et al., 2009). However, specific case studies on developing controllers for deformation processes using RL that learns directly by interacting with an FE environment are missing from the literature.

Machine learning (ML) can be divided into three main categories; supervised learning, unsupervised learning and RL. In short, unsupervised learning is used to find hidden patterns in data and supervised learning is used to learn to map inputs to outputs from labeled data examples. In contrast, RL algorithms do not need input/output example data and rather learn by interacting with an environment. An RL algorithm learns to map input environment observations to output actions that affect the environment by improving its actions according to a given reward system. Thus, RL algorithms are not limited by the quality of training data and have the possibility to discover solutions that are difficult for humans to conceive. However, the effectiveness of RL strongly depends on the quality of the reward system, which is designed by a human. Notable RL applications include self-driving cars, robotic control and algorithms that learn to play board games such as chess, shogi and Go on a superhuman level through self-play (Sallab et al., 2017; Kormushev et al., 2013; Silver et al., 2018). The literature includes studies on utilizing ML in a wide range of fields, including finance, medicine and engineering (Hutter et al., 2019; Shawi et al., 2019). However, there is a growing concern that studies on ML methods can be difficult to replicate, challenging their scientific credibility (Olorisade et al., 2017). This is due to both the inherent pseudorandom nature of training ML algorithms and the difficulty in interpreting their inner structures and architectures. Approaches to improving the reproducibility and interpretability of ML methods are currently being discussed in the

literature (Islam et al., 2017; Doshi-Velez & Kim, 2017; Ghanta et al., 2018). In the field of material science and engineering, investigating ML and its potential applications is especially promising due to the amount of available data and the need for increasingly complex models. Significant areas of study include accelerating the discovery of new materials, mapping material behavior to process variables and the prediction of phase diagrams and material properties (Mueller et al., 2016; Brunton & Kutz, 2019).

This work is focused on accelerating the hot deformation of the titanium aluminide alloy TNM-B1 (Ti-43.5Al-4Nb-1Mo-0.1B). This is an attractive material due to its high strength to weight ratio and high resistances to corrosion, temperature and creep (Clemens & Kestler, 2000; Clemens & Smarsly, 2011; Brotzu et al., 2018). TNM-B1 is particularly useful for the aerospace industry and has seen implementation in the form of forged turbine blades for commercial aircraft engines (Bewlay et al., 2016). However, the low workability of the alloy hinders wider implementation, as the material needs to be formed at low strain rates and high temperatures, resulting in high manufacturing costs for TNM-B1 forgings. Thus, accelerating the hot deformation of TNM-B1 can be a way to reduce costs, leading to an increased utilization of the alloy. This can be accomplished by exploiting the distinct hot deformation behavior of the material, which is characterized by a high initial peak stress followed by a strong softening behavior (i.e. flow stress reduction). Studies have indicated that the softening behavior of TiAl alloys during hot deformation is caused by dynamic recrystallization (DRX) (Beddoes et al., 1994; Millett et al., 1993; Kim & Boyer, 1991). By increasing the strain rate in accordance with the rate of softening, the deformation of the alloy can be accelerated. This was previously investigated experimentally by the authors (Stendal et al., 2021). This work was based on the Gurson-Tveergaard-Needleman damage model, which was first developed for steels in Tvergaard and Needleman (1984) and later extended to include DRX modeling in Bambach and Imran (2019). The model suggests that maximum strain rate while maintaining constant void nucleation rate can be achieved for titanium aluminides by achieving a constant stress state throughout deformation. The TNM-B1 alloy was deformed according to fixed accelerated strain rate profiles. However, the observed hot deformation behavior of the material was highly stochastic. The highest range in peak stresses for tests performed at the same conditions was around 40 MPa. Therefore, accelerating the deformation using the fixed strain rate profiles resulted in highly varied flow stress. Relatively hard samples underwent excessive acceleration, which led to increased flow stresses and potentially increased likelihood of workpiece damage. Relatively soft samples underwent moderate acceleration, which led to reduced flow stresses and unoptimized processing times. Thus, a process controller that adapts the deformation speed to the given workpiece behavior can optimize the processing time while avoiding excessive damage.

In this work, the development of a framework for interfacing an RL algorithm with an FE model in order to develop an artificial neural network (ANN) process controller is investigated. The aim is to accelerate the hot deformation of the titanium aluminide TNM-B1. Hot compression experiments were performed in order to determine the degree of variation in the deformation behavior of TNM-B1. The testing conditions used were the constant strain rates 0.0013, 0.005, 0.01 and 0.05 s^{-1} at a constant temperature of 1150 °C. A material model was fitted to the experimental data and used to predict flow stress surfaces. Synthetic variation was added to the surfaces using Gaussian rough surfaces in order to simulate variations in deformation behavior for the FE environment. The effect of geometry is important to consider, as local stresses, strains and strain rates can significantly vary from the nominal values depending on the shape of both the workpiece and the dies used. Ideally, the same RL setup can learn to adapt to different FE environments using different workpiece and die combinations. In order to investigate this, two separate FE environments were set up using the same RL setup; the hot compression of a cylinder workpiece between flat dies and the hot compression of a bone workpiece between flat dies. The bone geometry is typically used as the initial workpiece when forging turbine blades. For both environments, the RL algorithm adjusted the velocity of the die with the goal of achieving a constant stress state. This was carried out by rewarding the algorithm if the von Mises stress in chosen elements in the meshes of the two workpieces were within a set range of the initial peak stress throughout plastic deformation. The elements were chosen based on where the highest damage or the highest flow stress is likely to occur. The actions (i.e. outputs) of the RL algorithm were defined as die velocity adjustments and the observations (i.e. inputs) were defined as the displacement, force and velocity of the die. The observations were chosen as these are measurable in a physical compression process, theoretically making the ANN controller transferable to a real environment. The performances of the ANN controllers for the cylinder compression and bone compression environments and the behaviors of the process variables were tested and evaluated using 5 synthetic flow stress surfaces with global scale factors set to 0.9500, 1.0325, 1.1150, 1.1975 and 1.2800. In addition, the environments were tested using fixed acceleration profiles for comparison. Finally, the abilities of the ANN controllers to make decisions for stress surfaces with global scale factors outside the range used for training were evaluated.

Table 1 Chemical composition of the TNM-B1 ingot

	Al	Nb	Mo	B	O	Fe	Ni	C
at. %	43.7	4	1	0.1	0.161	0.027	0.008	0.038
wt %	28.65	9.15	2.36	0.026	0.063	0.037	0.012	0.011

Following this introduction, the “Methods and fundamentals” section provides an overview of the material and heat treatment used, the method used for performing the compression tests and a short introduction to the fundamentals of reinforcement learning. The “Model development” section is structured into two main parts. The first part presents the material model used, an alternative analytical method to constructing the accelerated strain rate profiles predicted by the reinforcement learning algorithm, as well as the development process for generating synthetic flow stress data for training. The second part provides a detailed description of the co-simulation setup used for adaptively interfacing reinforcement learning with a finite element model, the setup of the finite element models used, as well as a presentation of the reinforcement learning algorithm and reward function used. The results section provides and discusses the results of employing the developed reinforcement learning and finite element model co-simulation for the given cylinder compression and bone compression environments.

Methods and fundamentals

Material and heat treatment

The material used in this work was the titanium aluminide alloy TNM-B1. The composition is given in Table 1.

The TNM-B1 ingot was produced by GfE Metalle und Materialien GmbH (Nürnberg, Germany) using vacuum arc remelting (VAR) (Achtermann et al., 2009; Clemens & Mayer, 2013). The ingot was hot isostatically pressed (HIPed) in order to close casting microporosities, using the process parameters 1200 °C, 200 MPa, 4 h and a cooling rate of around − 8 K/min. The cylinder ingot had an initial height of around 150 mm and a diameter of around 49 mm and was cut to a height of around 95 mm. The ingot was subsequently eroded by Erocontur GmbH (Müncheberg, Germany) using electrical discharge machining (EDM) into smaller cylinders with heights of around 95 mm and diameters of around 7 mm.

The cylinders were subjected to a heat treatment which was developed and analyzed in an earlier collaborative study Eisentraut et al. (2019). The heat treatment was developed with the aim of improving the damage tolerance and hot workability of the material for accelerated deformation. This two-step heat treatment was performed using a batch furnace from Nabertherm GmbH (Lilienthal, Germany) at atmospheric conditions. The first step was conducted at 1300 °C for 1 h, the second step was 1100 °C for 5 h. Each step was followed by air cooling at atmospheric conditions down to room temperature. The heat treated TNM-B1 cylinders were machined to the final cylindrical test samples with a height of around 8 mm and a diameter of around 5 mm.

Compression tests

The hot compression tests were performed using a DIL805A/D/T deformation and quenching dilatometer from TA Instruments (Hüllhorst, Germany). All tests were performed at a constant temperature of 1150 °C. The samples were deformed at the constant strain rates 0.0013, 0.005, 0.01 and 0.05 s^{-1}. In order to investigate the variation in flow behavior, each strain rate condition was repeated 4 times. To protect the samples from oxidation, the compression tests were performed in an argon atmosphere. The dilatometer utilized Si_3N_4 punches with around 1 mm thick molybdenum plates between the punches and the sample. The samples were heated at around 10 K/s by induction. The samples were held at 1150 °C for 3 min in order to obtain a homogeneous microstructure and quasi-isothermal conditions, before being deformed to a true strain of around 0.8. Finally, the samples were cooled at around − 150 K/s using argon gas.

Reinforcement learning fundamentals

This section covers the basic fundamentals and terminology of the reinforcement learning (RL) framework and the RL algorithm used in this work. RL involves an algorithm that performs actions that affect an environment. The algorithm receives observations from the environment and learns to improve its actions according to a given notion of reward. Basically, the actions are the outputs and the observations are the inputs of the controller. The algorithm attempts to maximize the reward through an iterative process of trial and error, where the action space is searched for decisions that lead to increased rewards. Each training iteration is called an episode. For further reading on RL fundamentals, see Sutton and Barto (2018).

Figure 1 illustrates the architecture of the RL and FE co-simulation used in this work. The agent consists of the learning algorithm and the policy, which is the decision-making function that performs the actions. The environment is the FE simulation, which the agent communicates with through observations, actions and rewards. The actions are sent to the environment from the agent. The observations are sent to the agent from the environment. The rewards are calculated by a reward function and are sent to the agent. Typically, the reward is a scalar value and provides the learning algorithm with the degree to which the environment is in a desirable state. During training, the RL algorithm updates the policy based on the performed actions, the environment observations and the amount of collected reward.

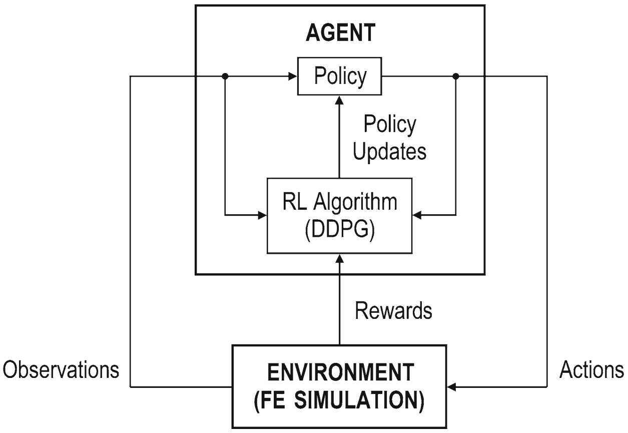

Fig. 1 Schematic of the reinforcement learning (RL) and finite element (FE) co-simulation using the deep deterministic policy gradients (DDPG) RL algorithm

The RL algorithm used in this work was the policy-gradient actor-critic algorithm deep deterministic policy gradients (DDPG). This algorithm is based on the work done on deterministic policy gradients (Silver et al., 2014). It was developed by Google DeepMind with the aim of handling continuous action spaces (Lillicrap et al., 2016). Other reinforcement learning algorithms that can handle continuous observation and action spaces could also have been considered, such as asynchronous advantage actor critic (A3C) or proximal policy optimization (PPO). Traditionally, RL has been limited to discrete action spaces. The challenge of tackling continuous ones has been an important problem to solve in the field. The use of continuous actions spaces is essential for a range of real-world applications such as robotic control (Mnih et al., 2015).

The actor-critic architecture used in the DDPG algorithm is illustrated in Fig. 2. The actor is the policy function, which maps the given environment state (s) to actions (a). The critic predicts a Q-value (Q) based on the state (s) and action (a). The Q stands for the quality of a performed action and represents the long-term potential reward starting from a given state and executing a specific policy. The Q-value is used to update the actor and the critic itself. Policy-gradient algorithms works by estimating a gradient of the expected reward of a given policy. The policy is then updated in the direction of the gradient in order to increase the probability of performing desirable actions over time.

The artificial neural network (ANN) machine learning framework was used for both the actor and the critic, as illustrated in Fig. 2. In short, an ANN is a network of interconnected linear equations with nonlinearity introduced using activation functions (Anderson, 1995). One linear equation

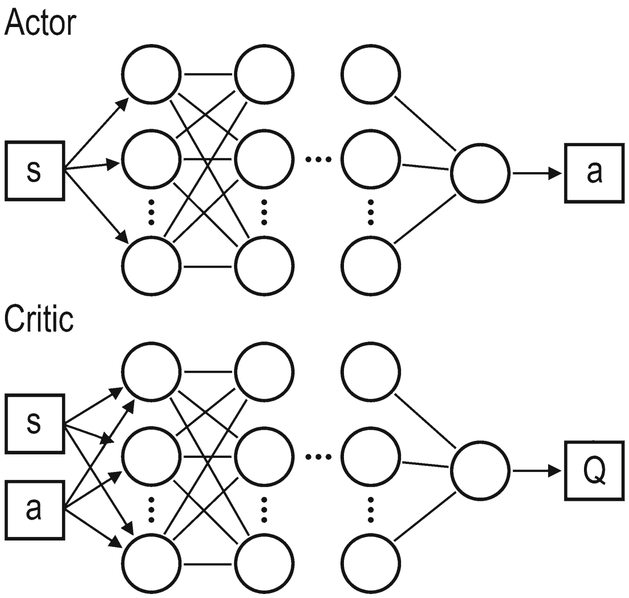

Fig. 2 Actor-critic artificial neural network (ANN) architecture; s: environment state, a: actions, Q: value function. Both the actor and critic are represented as ANNs

and activation function pair is called a neuron. The neurons are arranged in layers; an input and an output layer with so-called hidden layers in between. If the number of hidden layers exceed 1, the ANN is referred to as deep. The universal approximation theorem in mathematics states that a feed-forward ANN with a single hidden layer containing a finite number of neurons can approximate continuous functions on compact subsets of the real coordinate space, under mild assumptions on the activation function (Cybenko, 1989). Thus, an ANN can be described as a universal function approximator (Hornik et al., 1989).

Model development

In this section, the development of the material model, defining fixed acceleration profiles, generating synthetic flow stress data and the development of the RL and FE co-simulation are discussed.

Material model

The material model was used to predict the flow stress behavior of the TNM-B1 alloy. The model is a hybrid model consisting of an artificial neural network (ANN) and a physics-based phenomenological model (PM). It was developed and presented in a previous work by the authors (Stendal et al., 2019). The PM was developed for steel by Cingara and McQueen (1992) and later adapted to titanium aluminides by Bambach et al. (2016). In Table 2, the model equations of the PM are displayed. In short, characteristic points along the experimental flow curves are extracted. These points are the critical stress (σ_c), which marks the onset of dynamic recrystallization (DRX), the peak stress (σ_p) and the steady state stress (σ_{ss}). The points are expressed as functions of the Zener Hollomon parameter Z (Eq. 1) and used to fit the parameters of the model equations to the experimental data "Experimental compression tests" section using regression analysis.

The hybrid model is illustrated in Fig. 3. The ANN component of the hybrid model improved the fit of the original PM to the experimental flow curves. Basically, the ANN was trained on the flow curve data with the aim of predicting the positions of the characteristic points as functions of temperature and strain rate. These predictions subsequently provided the PM with a wider range of input characteristic points, which resulted in a closer fit to the experimental flow curves.

Table 2 Phenomenological model (MP) equations for used for TNM-B1

ZHP	$Z = \dot{\varepsilon} \cdot exp(\frac{Q_w}{RT})$	(1)
Strain hardening	$\sigma(\varepsilon) = \sigma_p\left[\frac{\varepsilon}{\varepsilon_p} exp\left(1 - \frac{\varepsilon}{\varepsilon_p}\right)\right]^C$	(2)
Critical strain	$\varepsilon_{cr} = \alpha\varepsilon_p$	(3)
Peak strain	$\varepsilon_p = a_1 \cdot d_0^{a_2} \cdot Z^{a_3}$	(4)
Steady state strain	$\varepsilon_{ss} = e_1 \cdot \varepsilon_m + e_2 \cdot d_0^{e_3} \cdot Z^{e_4}$	(5)
Peak stress	$sinh(f_3 \cdot \sigma_p) = f_1 \cdot Z^{f_2}$	(6)
Steady state stress	$sinh(h_3 \cdot \sigma_p) = h_1 \cdot Z^{h_2}$	(7)
DRX grain size	$d_{DRX}(\gamma, \beta) = b_1(\gamma, \beta) \cdot Z^{b_2(\gamma,\beta)}$	(8)
DRX kinetics	$X_{DRX}(\gamma, \beta) = 1 - exp(k(\gamma, \beta)\left(\frac{\varepsilon-\varepsilon_{cr}}{\varepsilon_{ss}-\varepsilon_{cr}}\right)^{q(\gamma,\beta)}$	(9)
Flow stress	$\sigma_y = \begin{cases} \sigma_0 & if\ \varepsilon < \varepsilon_{cr} \\ \left(1 - (X_\gamma + X_\beta)\right)\sigma_0 + (X_\gamma + X_\beta)\sigma_1 & if\ \varepsilon > \varepsilon_{cr} \end{cases}$	(10)

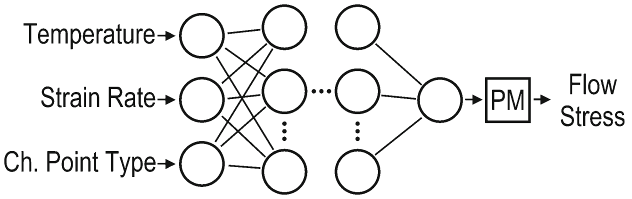

Fig. 3 Schematic of the hybrid model, consisting of the ANN and the phenomenological model (PM)

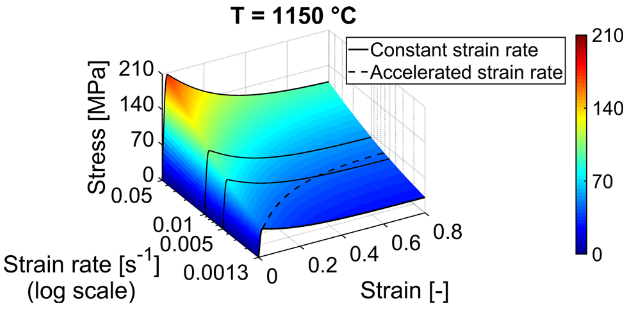

Fig. 4 Flow stress surface predicted by the hybrid material model for 0 to 0.8 strain and 0.0013 to 0.05 s^{-1} at 1150 °C. The fixed accelerated strain rate profile is shown in the dotted black line plot

Fixed acceleration profiles

The hybrid material model was used to predict a flow stress surface, as shown in Fig. 4, based on the experimental data "Experimental compression tests" section. The surface was generated using 237 logarithmically spaced flow curves from 0.0013 to 0.05 s^{-1} at 1150 °C. Each flow curve has 1000 evenly spaced data points from 0 to 0.8 strain. The solid black line plots in Fig. 4 represents the flow curves at the constant strain rates from where the experimental data were obtained (0.0013, 0.005, 0.01 and 0.05 s^{-1}).

The softening behavior of TNM-B1 allows increasing the strain rate during deformation without exceeding the initial peak stress. This work is focused on the strain rate profile that maintains the initial peak stress constant during plastic deformation, with a starting strain rate of 0.0013 s^{-1}. This accelerated strain rate profile is represented by the dotted black line plot in Fig. 4 and was obtained using a numerical search function to find the strain and strain rate coordinates that maintain the initial peak stress at 0.0013 s^{-1} within ±0.2 MPa. The profile follows an S-shaped curve and thus can be fitted to the logistic function (Eq. 11):

$$\dot{\varepsilon}(\varepsilon) = \frac{L}{1 + e^{-k(\varepsilon - \varepsilon_0)}} \tag{11}$$

where ε is the strain, $\dot{\varepsilon}$ is the strain rate, L is the curves maximum value, k is the logistic growth rate or curve steepness and ε_0 is the strain value at the curves midpoint. The parameters were determined using nonlinear regression. Figure 5a displays the accelerated strain rate profile extracted from the material model and the fitted logistic function. Implementing the accelerated profile in forging applications and FE simulations requires the strain rate profile to be expressed as velocity as a function of time. Using $\dot{\varepsilon} = \dot{\varepsilon}(\varepsilon), d\varepsilon/dt = \dot{\varepsilon}(\varepsilon)$ and $dt = d\varepsilon/\dot{\varepsilon}(\varepsilon)$, the time as a function of strain with the fitted logistic function substituted for the strain rate as a function of strain can be expressed in Eq. (12):

$$t(\varepsilon) = \int \frac{1}{\dot{\varepsilon}(\varepsilon)} d\varepsilon = \frac{\varepsilon - \frac{e^{k(\varepsilon_0 - \varepsilon)}}{k}}{L} + constant \tag{12}$$

where t is the time. The constant can be found by setting t(0) = 0. The accelerated strain rate profile can then be converted to an accelerated velocity profile first by converting the true strain values from the material model to engineering strain values and finally by converting engineering strain rate to velocity according to the following steps: $\varepsilon_{engineering} = 1 - 1/\varepsilon_{true}$, $\dot{\varepsilon}_{engineering} = d\varepsilon_{engineering}/dt$ and $v = \dot{\varepsilon}_{engineering} \cdot h_0$, where ε_{true} is the true strain, $\varepsilon_{engineering}$ is the engineering strain, v is the velocity and h_0 is the initial workpiece height.

Figure 5b displays the accelerated velocity profile for an initial strain rate of 0.0013 s^{-1} as well as the velocity profile for a constant 0.0013 s^{-1} as functions of time for a cylinder geometry with a height of 8 mm. Using the 0.0013 s^{-1} constant strain rate profile, the theoretical processing time is around 615 s. Using the accelerated profile with a starting strain rate of 0.0013 s^{-1}, the theoretical processing time can be reduced to around 235 s. Thus, a reduction of around 62 % can be achieved.

The bone geometry investigated in this work is illustrated in Fig. 6. The heights of both bases were chosen to be 44 mm. As the bone geometry is compressed between flat dies, the height of the shaft was chosen to be 20 mm in order to reach around 0.8 global true strain once the dies make contact with the shaft. A final stroke height of 9 mm was chosen in order to reach around 1.6 global true strain at the end of compression. The acceleration profile calculated for the cylinder geometry (Fig. 5) does not consider the switch from deforming the bases to deforming the shaft at around 0.8 global strain. Therefore, accelerating the compression of the bone geometry using the same profile will lead to the shaft undergoing increased flow stresses compared to the bases. This can result in increased likelihood of workpiece damage or die fracture. In order to avoid this, both the bases and the shaft can be deformed according to the acceleration profile by constructing an adapted profile with a reset at around 0.8 global true strain. A sigmoid transition (Eq. 13) was utilized in order to construct a smooth transition between two discontinuous acceleration profiles at the switch between deforming only the bases to deforming the whole bone geometry:

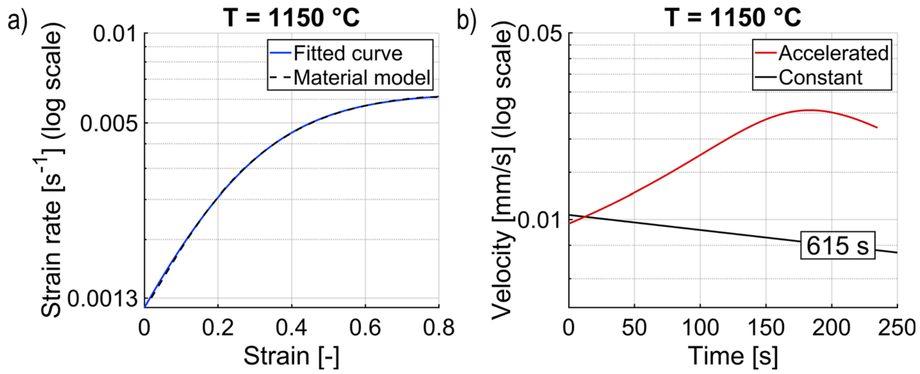

Fig. 5 **a** accelerated strain rate profile extracted from the hybrid material model and the fitted logistic function, **b** accelerated velocity profile starting at 0.0013 s^{-1} and the velocity profile corresponding to a constant 0.0013 s^{-1} for a cylinder geometry with a height of 8 mm

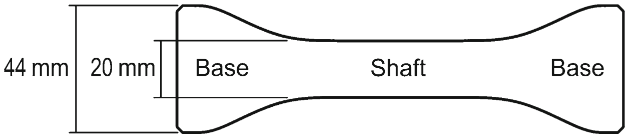

Fig. 6 Illustration of the bone geometry used in this work

$$\dot{\varepsilon}(\varepsilon) = \big(1 - \sigma(\varepsilon)\big)\dot{\varepsilon}_1(\varepsilon) + \sigma(\varepsilon)\dot{\varepsilon}_2(\varepsilon - \varepsilon_s) \tag{13}$$

where $\dot{\varepsilon}(\varepsilon)$ is the constructed acceleration profile for the bone geometry, $\sigma(\varepsilon)$ is the sigmoid function, $\dot{\varepsilon}_1(\varepsilon)$ is the logistic function before the switch, $\dot{\varepsilon}_2(\varepsilon - \varepsilon_s)$ is the logistic function after the switch and ε_s is the strain value at the switch (set to 0.8). Following the logic of this construction, the sigmoid function and logistic functions can be substituted, resulting in an expanded form of the acceleration profile for the bone geometry (Eq. 14):

$$\begin{aligned}\dot{\varepsilon}(\varepsilon) = {} & \left(1 - \frac{1}{1 + e^{-k_\sigma(\varepsilon - \varepsilon_\sigma)}}\right)\frac{L}{1 + e^{-k(\varepsilon - \varepsilon_0)}} \\ & + \frac{1}{1 + e^{-k_\sigma(\varepsilon - \varepsilon_\sigma)}}\,\frac{L}{1 + e^{-k(\varepsilon - \varepsilon_0 - \varepsilon_s)}}\end{aligned} \tag{14}$$

where L, k and ε_0 are the parameters of the logistic function. k_σ is the steepness of the sigmoid transition between the profiles (set to 50). This value can be set depending on the ability of the physical equipment to make velocity adjustments. ε_σ is the strain value of the midpoint of the sigmoid transition. This was set to 0.7 in order for the strain rate to decrease to the initial strain rate of around 0.0013 s^{-1} before the strain value of 0.8 is reached. Using this method, the logistic function can be nested to include an arbitrary number of resets with smooth sigmoid transitions. Using the same procedure as in Eq. (12), strain rate as a function of strain can be converted to time as a function of strain as stated in Eq. (15):

$$t(\varepsilon) = \int \frac{1}{\left(1 - \frac{1}{1+e^{-k_\sigma(\varepsilon-\varepsilon_\sigma)}}\right)\frac{L}{1+e^{-k(\varepsilon-\varepsilon_0)}} + \frac{1}{1+e^{-k_\sigma(\varepsilon-\varepsilon_\sigma)}}\,\frac{L}{1+e^{-k(\varepsilon-\varepsilon_0-\varepsilon_s)}}}\,d\varepsilon \tag{15}$$

This integral has no analytical solution. However, it can be calculated using a numerical method for the strain interval of 0 to 1.6. The accelerated velocity profile for the bone geometry with a smooth sigmoid transition at around 0.8 global true strain was subsequently calculated using the steps: $\varepsilon_{engineering} = 1 - 1/\varepsilon_{true}$, $\dot{\varepsilon}_{engineering} = d\varepsilon_{engineering}/dt$ and $v = \dot{\varepsilon}_{engineering} \cdot h_0$.

Figure 7 displays the accelerated strain rate and velocity profiles for the bone geometry. The accelerated profiles without a reset at 0.8 true strain are shown in the dotted black line plots for comparison. Without the sigmoid transition, the theoretical processing time is around 360 s. Using the reset, the theoretical processing time is increased to around 520 s, giving an increase of around 44%. However, the likelihood of workpiece damage or die fracture can be reduced.

Generating synthetic flow stress data

The quality and applicability of the ANN controller developed by the RL agent strongly depends how closely the FE environment represents the real-world process. Therefore, providing the FE model with realistic material behavior is important. The experimental flow curves cannot be used directly, as they are not correlated to each other. In addition, the collected dataset is too small for the many training episodes required to converge on a desirable solution. The material model cannot be used directly, as providing the environment with identical material behavior in every episode can lead to a phenomenon called overfitting. In statistics, overfitting occurs when model predictions correspond too closely to a dataset, making the model less able to make generalized predictions on additional data. In RL, overfitting involves an agent in a sense "memorizing" sequences of actions by exploiting the determinism of the environment. Adding stochastic behavior to the environment can be an effective method of reducing memorization, leading to the development of a controller that performs better for environment states outside the ranges used for training. However, using stochastic environments cannot fully prevent overfitting in RL, as was shown by Zhang et al. (2018).

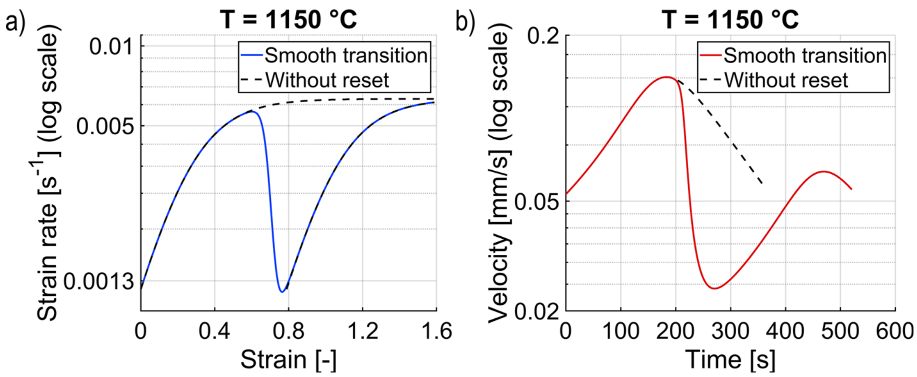

Fig. 7 **a** Accelerated strain rate profile adapted to the bone geometry, **b** accelerated velocity profile adapted to the bone geometry. The dotted black line plots in both **a** and **b** show the acceleration profiles without a sigmoid transition

In order to provide the FE environment with realistic stochastic material behavior as well as to reduce overfitting, synthetic flow stress surfaces were generated based on the assumptions made on the range of possible deformation behaviors from the experimental flow curves "Experimental compression tests" section. A unique stress surface was generated for every training episode, simulating the possible variations in deformation behavior for each workpiece. In order to extrapolate the observations made from the 2D experimental curves to a 3D surface, it was assumed that a given workpiece exhibits the same relative deformability independent of strain rate. For instance, a workpiece that displays relatively low deformability at low strain rates will display the same relatively low deformability at high strain rates. In order to simulate this, the entire stress surface predicted by the material model was scaled in the stress direction using a variable global scale factor.

Figure 8 displays two flow stress surfaces predicted by the material model. The experimental flow curves are shown in the solid black line plots. In Fig. 8a, a global scale factor of 0.9500 was applied in order to approximately intersect with the lowest peak stress in the experimental data. In Fig. 8b, a global scale factor of 1.2800 was applied to approximately intersect with the highest experimental peak stress. Thus, the range of possible experimental peak stresses can be simulated by varying the global scale factor between 0.9500 and 1.2800.

The experimental flow curves of TNM-B1 display roughnesses and oscillations of varying scales "Experimental compression tests" section. This can be simulated by convoluting the material model stress surface with a pseudorandom rough surface. However, the roughness observed in the experimental flow curves demonstrates correlation rather than random noise. For this reason, the roughness and oscillations in the deformation behavior was simulated using pseudorandom Gaussian rough surfaces with correlation. The method used was presented by Garcia and Stoll (1984) and involves convoluting a distribution of uncorrelated pseudorandom numbers with a Gaussian filter to achieve correlation over the distribution. The convolution was performed using the fast-Fourier-transform (FFT) algorithm. Equation (16) describes the Gaussian filter:

$$G = exp\left[-\frac{x^2+y^2}{\frac{cl^2}{2}}\right] \tag{16}$$

where cl is the correlation length, which determines the filter frequency width. A larger cl corresponds to smaller spatial variations over the surface and vice versa. The cl parameter and the scale factor of the Gaussian rough surface itself can be adjusted to generate surfaces with varying heights and frequency widths. To simulate the different scales of the roughnesses and oscillations observed in the experimental data, 3 layers of pseudorandom Gaussian rough surfaces were convoluted with the material model stress surface.

Figure 9 displays generated examples of the 3 different layers of Gaussian rough surfaces. As discussed in "Experimental compression tests" section, the softening behavior can be independent of the peak stress. Therefore, layer 1 was designed to decouple the peak stress from the softening rate. This was achieved by convoluting layer 1 with an S-shaped sigmoid surface, where the scaling is around 1 until the peak stress is reached and variable beyond the peak stress. Layer 2 was designed to simulate the large-scale roughnesses and oscillations observed in the experimental data and layer 3 was designed to simulate the small-scale roughnesses. The cl parameters, sigmoid parameters and Gaussian rough surface scale factors were found through a process of trial and error, in order to produce synthetic flow curves that were observed to correlate with the experimental data. The cl values were set to 1, 6 and 30 for layers 1, 2 and 3, respectively. The scale factor ranges for each layer were set to 0.90 to 1.10 for layer 1, 0.96 to 1.04 for layer 2 and 0.99 to 1.01 for layer 3. Equation (17) describes how the final synthetic stress surfaces were generated by convoluting the 3 layers of Gaussian rough surfaces with the material model stress surface and applying a global scale factor between 0.9500 and 1.2800:

$$\begin{aligned} S_{synthetic} = {} & S_{mm} \cdot \lambda_{global} \cdot S_{G1} \cdot \lambda_{G1} \cdot S_{G2} \cdot \lambda_{G2} \\ & \cdot S_{G3} \cdot \lambda_{G3} \end{aligned} \tag{17}$$

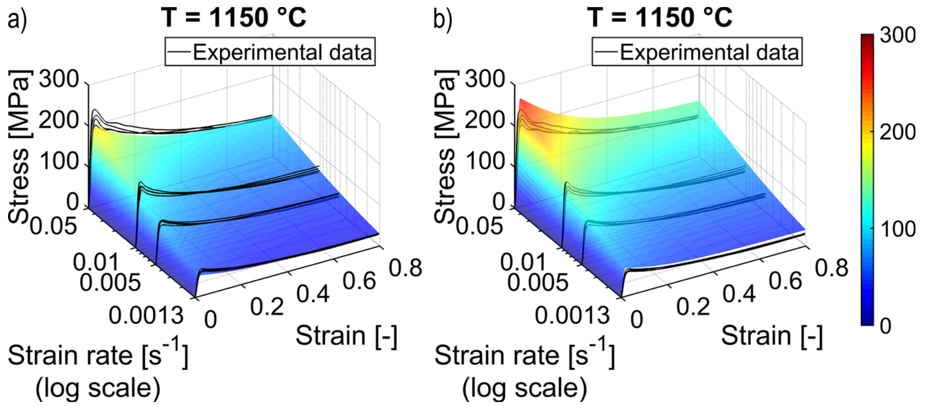

Fig. 8 Material model stress surfaces using: **a** a global scale factor of 0.9500, **b** a global scale factor of 1.2800. The experimental flow curves are displayed in the solid black line plots

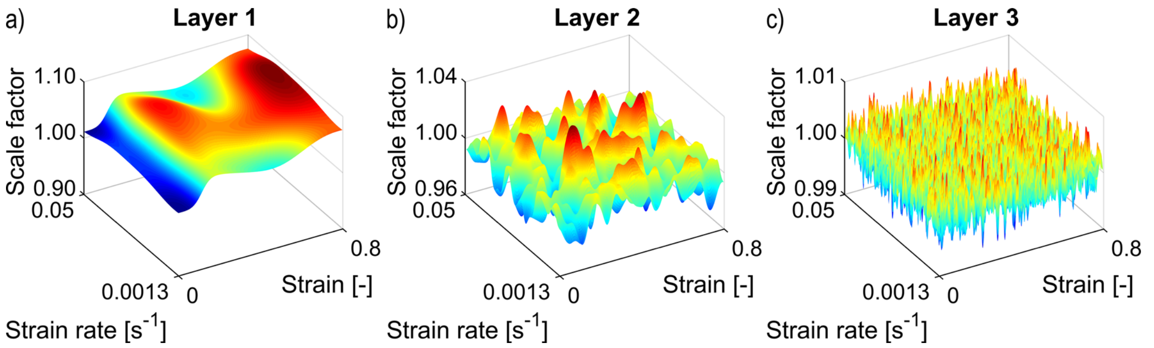

Fig. 9 **a** Layer 1, designed to decouple peak stress and softening rate, **b** layer 2, designed to simulate large-scale roughnesses and oscillations, **c** layer 3, designed to simulate small-scale roughnesses

where $S_{synthetic}$ is the synthetic stress surface, S_{mm} is the material model stress surface, λ_{global} is the global scale factor, S_{G1}, S_{G2} and S_{G3} are the Gaussian rough surface layers and λ_{G1}, λ_{G2} and λ_{G3} are their respective scale factors. In the final synthetic stress surface generator, the global scale factor, the Gaussian rough surface scale factors, the Gaussian rough surface parameters and the sigmoid parameters of layer 1 were set to pseudorandom numbers varying between the stated limits in order to produce a unique stress surface for each training episode.

Figure 10 displays the experimental flow curves and synthetic flow curves at 0.0013, 0.005, 0.01 and 0.05 s^{-1}. The synthetic flow curves are from 4 synthetic stress surfaces generated using pseudorandom parameters and the set global scale factors of 0.9500, 1.0600, 1.1700 and 1.2800. The variations in peak stresses, softening rates, roughnesses and oscillations correlate with the experimental data. However, the range of possible synthetic curves is slightly wider than the experimental ones, which can lead to the development of a more conservative ANN controller. A closer correlation between the experimental and synthetic data can be achieved by improving the fit of the original material model stress surface.

Figure 11 displays the histograms for the experimental and synthetic data distributions. The synthetic data were obtained from 1000 synthetic stress surfaces generated using pseudorandom parameters. The material model stress surface is shown in the vertical black line plots. The synthetic data approach a normal distribution with a wider range compared to the experimental data. Furthermore, the synthetic distribution is slightly biased in the positive direction as the global scale factors range from 0.9500 to 1.2800. The experimental and synthetic data distributions are similar. However, a histogram does not provide a quantitative measure of the quality of the synthetic data. Generating synthetic data for use in ML is a recent field of study (Choi et al., 2017; Esteban et al., 2017; Xie et al., 2018). Therefore, reliable methods of measuring the quality of synthetic data are still being discussed in the literature (Jordon et al., 2018). Kaloskampis et al. (2019) and Beaulieu-Jones et al. (2019) suggested comparing pairwise Pearson correlations (Pearson, 1901) in the synthetic dataset to the ones in the real dataset, as displayed to Eq. (18):

$$\Delta Pcorrelation = Pcorrelation_{real} - Pcorrelation_{synthetic} \tag{18}$$

where $Pcorrelation_{real}$ is the pairwise Pearson correlation in the real data and $Pcorrelation_{synthetic}$ is the pairwise Pearson correlation in the synthetic data. The difference

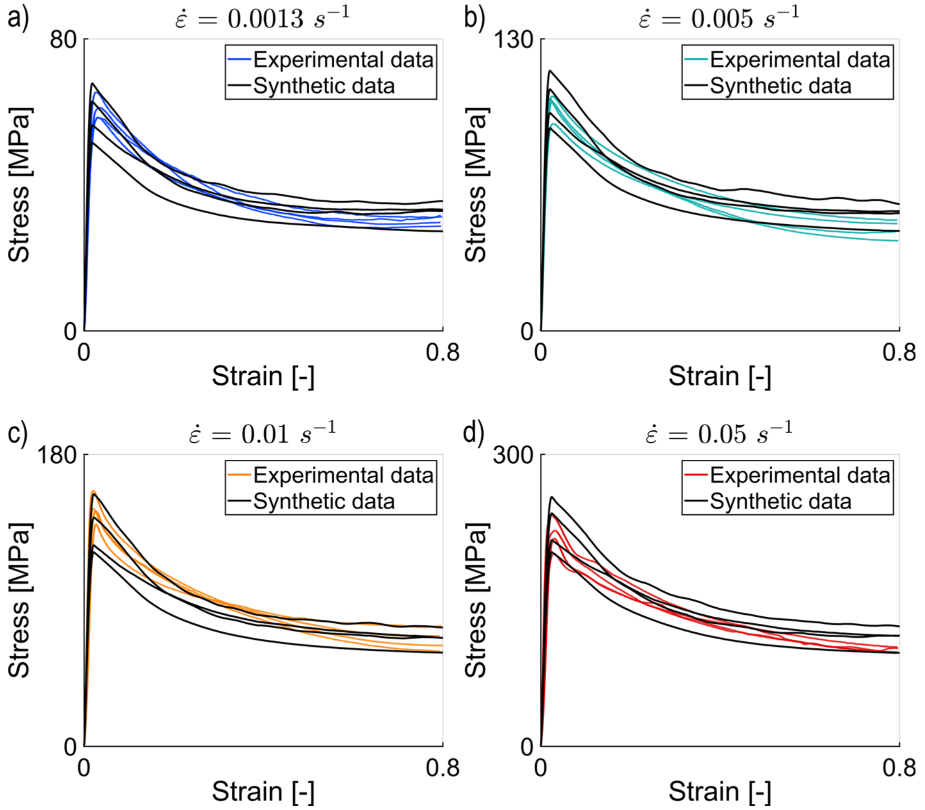

Fig. 10 Experimental and synthetic flow curves for **a** 0.0013 s^{-1}, **b** 0.005 s^{-1}, **c** 0.01 s^{-1} and **d** 0.05 s^{-1}

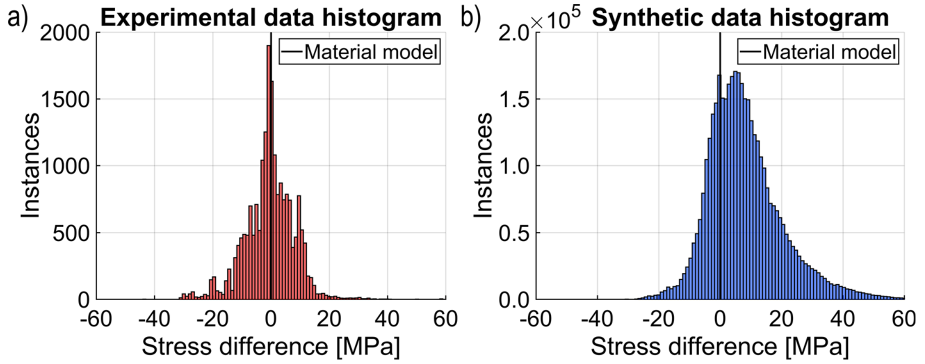

Fig. 11 **a** Histogram of the experimental data and **b** histogram of the synthetic data from 1000 generated surfaces. Both histograms are compared to the original material model stress surface displayed in the vertical black solid line plots

between these values ($\Delta Pcorrelation$) is used to determine if the relationship between the variables in the real data is preserved in the synthetic data. A value of $\Delta Pcorrelation$ closer to zero indicates higher quality synthetic data. In this work, the $\Delta Pcorrelation$ was calculated to an average of around 0.02 by comparing the experimental flow curves to 1000 sets of 4 generated synthetic flow curves for each strain rate condition.

Co-simulation setup

The RL and FE co-simulation interface was set up in MATLAB. Approaches to linking MATLAB with FE programs have previously been discussed in the literature (Orszulik & Gabbert, 2016; Almandoz et al., 2012). Alternatively, the framework can be set up licence free using Python, as the libraries stable-baselines3 and gymnasium are streamlined for developing and interfacing RL algorithms with custom built environments. In this work, the FE program used was LS Dyna, as MATLAB can communicate with it directly by manipulating and executing keyword files (.k) and command files (.cfile). The FE program Abaqus is also a good candidate, as it can be controlled directly through the manipulation of text files. The co-simulation was performed using an average Intel i5 without GPU acceleration and the training times were measured using the same hardware for all conditions. The method used for setting up the FE part of the co-simulation was inspired by a method proposed by Strano et al., which involved dividing tube hydroforming simulations into intervals with the goal of adjusting the loading

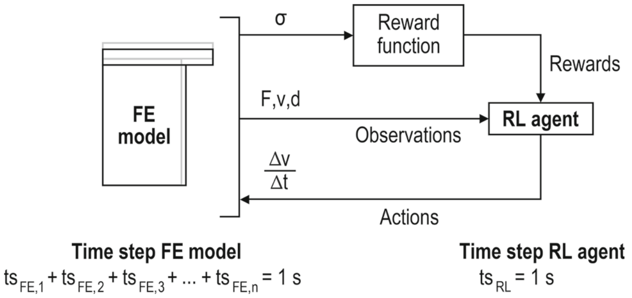

Fig. 12 Schematic illustrating the reinforcement learning (RL) and finite element (FE) co-simulation interface and the calculation of time steps. σ: stress, F: force, v: velocity, d: displacement, $\Delta v/\Delta t$: velocity adjustments

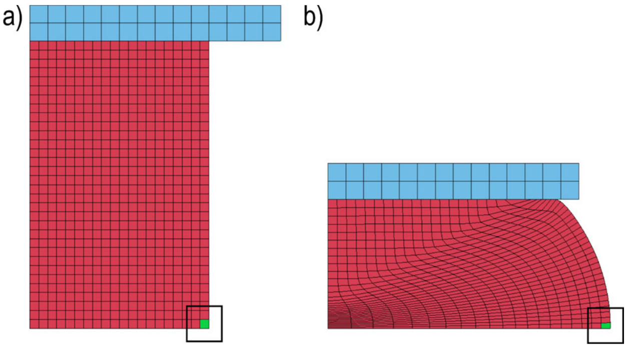

Fig. 13 FE model of a 1/4 axisymmetric cylinder compression between flat dies; **a** initial mesh and **b** fully deformed mesh. The chosen element is displayed in green (Color figure online)

path within the same simulation run in order to avoid the onset and growth of defects (Strano et al., 2001).

As illustrated in Fig. 12, the communication between the RL agent and the FE simulation was conducted through 6 channels; σ: von Mises stress from a chosen element in the workpiece mesh, F: die force in the z direction, v: die velocity in the z direction, d: die displacement in the z direction and finally $\Delta v/\Delta t$: change in die velocity over time in the z direction. In the RL framework, F, v and d represents the observations, $\Delta v/\Delta t$ represents the actions and σ is used to calculate the reward for the RL agent. Data transmission was conducted by reading and writing text files. This method was computationally expensive. Therefore, alternative methods of interfacing are suggested for future research in order to reduce training times.

The FE simulation and RL agent was set to communicate at 1 s intervals. This communication interval can be set based on how often the physical press is able to update its ram velocity. For a servo-driven forging press, assuming 1 update per second is a reasonable starting point. Between each communication interval, the values of F, v and d are sent from the FE simulation to the RL agent. σ is sent to the reward function, which computes a reward that is sent to the agent. The agent subsequently predicts a $\Delta v/\Delta t$ which is added to the constant die velocity for the next communication interval. In order to match the communication intervals of the FE simulation and the RL agent, the time step of the agent was set to 1 s and an adaptive time step was utilized for the FE simulation. The adaptive time step was used as using a fixed time step can lead to error terminations if the amount of deformation exceeds the element size in a single time step. The adaptive length of each time step in the FE simulation interval was calculated using Eq. (19):

$$ts_{FE} = \begin{cases} \frac{v_{initial}}{v}, & \text{if } v \geq v_{initial} \\ 1, & \text{if } v < v_{initial} \end{cases} \tag{19}$$

where ts_{FE} is the FE time step, $v_{initial}$ is the initial die velocity and v is the die velocity. Thus, the time step length is set to 1 s at the initial velocity and decreases with increasing die velocity. If v falls below $v_{initial}$, ts_{FE} is set to 1 s. In LS Dyna, the termination time of each interval was set to 1 s. Between each communication interval, the simulation results were opened and the variable values were extracted. Subsequently, the current state of the simulation is saved and updated with a predicted $\Delta v/\Delta t$ value added to the die velocity and a time step length for the next communication interval. σ and F were the average values calculated from all the time steps of the interval. v and d were the values from the last time step of the interval.

Finite element model setup

Two FE environments were investigated in this work; the compression of a cylinder geometry between flat dies and the compression of a bone geometry between flat dies. Implicit analysis was used for both environments. A constant temperature of 1150 °C was used. Thermal analysis was not included. The dies were modeled as rigid bodies. The Tabulated Johnson Cook card was used to input the generated synthetic stress surfaces. In both the cylinder and the bone workpiece geometries, elements were chosen from where σ was extracted. The behavior of the final ANN controller strongly depends on the choice of elements. For the cylinder geometry, the element was selected based on the region where most of the damage was observed in the tested samples. For the bone geometry, the elements were chosen to represent the deformation behavior of the base and the shaft separately. For forging simulations, adaptive remeshing is often used as the elements can become significantly deformed. However, LS Dyna does not currently support point tracking. Therefore, it was not possible to utilize remeshing in this work.

Figure 13 displays the FE model of the cylinder compression between flat dies. The die mesh is marked in blue, the workpiece mesh is marked in red and the chosen element from where σ was extracted is marked in green. The cylinder compression was simplified to a 1/4 2D axisymmetric model in order to reduce the simulation time and thereby the training time. The workpiece mesh dimensions were a height of 4 mm and a width of 2.5 mm. The workpiece mesh consisted of 640 square 2D shell elements with 4 through shell thickness integration points and Belytschko-Bindeman hourglass element control (Belytschko & Bindeman, 1993). The contact modeling between the workpiece and the die was 2D automatic surface to surface with the coefficient of friction set to 0.1. The initial die velocity in the z direction was set to 0.0052 mm/s as this corresponds to 0.0013 s^{-1} for an initial height of 4 mm.

Figure 14 displays the FE model of the bone compression between flat dies. The die mesh is marked in blue, the workpiece mesh is marked in red and the chosen groups of elements from where σ was extracted is marked in green. The stress used for σ was calculated from the average von Mises stress in the group. Only the highest value of the two groups was used for σ. The bone compression was simplified to a 1/8 3D model in order to reduce simulation time. The workpiece mesh dimensions were a base height of 22 mm, a shaft height of 10 mm and a total length of 75.5 mm. The workpiece mesh consisted of 60,564 constant stress tetrahedral solid elements. The contact modeling between the workpiece and the die was automatic surface to surface with the coefficient of friction set to 0.1. The initial die velocity in the z direction was set to 0.0286 mm/s, as this corresponds to 0.0013 s^{-1} for an initial height of 22 mm. As can be observed, the deformations in the base part are unrealistically high. Using a more complex die can result in a more realistic deformation of the workpiece. However, this can result in significantly increased simulation times and is beyond the scope of this study.

Reinforcement learning setup

The basic fundamentals and terminology of reinforcement learning (RL) and the RL algorithm used in this work (DDPG) are discussed in "Reinforcement learning fundamentals section". Identical RL parameters and ANN architectures were used for both the investigated cylinder compression and bone compression FE environments. The DDPG agent parameters are given in Table 3. These were determined iteratively through a process of trial and error where the RL agents resulting from training were evaluated based on performance and training time. Basically, the parameters that lead to agents which showed an ability to generalize on the problems and to achieve high average rewards without excessively increasing the training times were chosen. During each training episode, the agent updates the actor and critic networks using experiences randomly sampled from an experience buffer. The mini batch size determines the number of experience samples. Larger batch sizes can reduce the training variance at the cost of computing time. The learning rate or step size parameter determines the rate at which the agent overwrites old behavior with new behavior. A lower learning rate can lead to longer training times, while a higher learning rate can lead to sub-optimal results or divergence. In general, lower learning rates are typically used for stochastic environments. A discount factor is applied to future rewards during training and determines their importance. A discount factor closer to 1 can increase the ability of the agent to pursue more long-term rewards. A smoothing method of updating the target actor and critic parameters was used. The Ornstein-Uhlenbeck process (Lemons & Gythiel, 1997; Uhlenbeck & Ornstein, 1930) was used as the noise model for the DDPG algorithm.

Table 3 DDPG agent parameters

Policy	MlpPolicy
Action noise model	Ornstein-Uhlenbeck
Action variance	0.1
Action variance decay rate	10^{-6}
Experience buffer length	10^{6}
Soft update coefficient	0.003
Mini batch size	256
Learning rate	0.001
Discount factor	0.99
Smoothing factor	0.001

In the RL agent, both the actor (i.e. policy function) and the critic (i.e. value function) are approximated by artificial neural networks (ANNs). The actor ANN consisted of an input layer with 3 neurons, 3 hidden layers and an output layer with 1 neuron. The critic ANN consisted of an observation path, an action path and a common path. The observation path consisted of an input layer with 3 neurons and 2 hidden layers and the action path consisted of an input layer with 1 neuron and 1 hidden layer. The observation and action paths were connected to the common path via an addition layer. The common path consisted of 2 hidden layers and an output layer with 1 neuron. All layers were fully connected. All hidden layers of both networks consisted of 50 neurons. The rectified linear activation function was used for both the actor and the critic. The observations (F, v and d) from the FE simulation were normalized by linearly scaling them to values roughly between 0 and 1 before being connected to the input layers of the actor and the observation path of the critic. Normalizing the scale of the input values can stabilize and accelerate the learning process. In addition, normalization can prevent a single observation from dominating the ANN predictions. After the actor output layer, the hyperbolic tangent (tanh) function was used to scale the action output to values between − 1 and 1. The output actions of the agent were linearly scaled

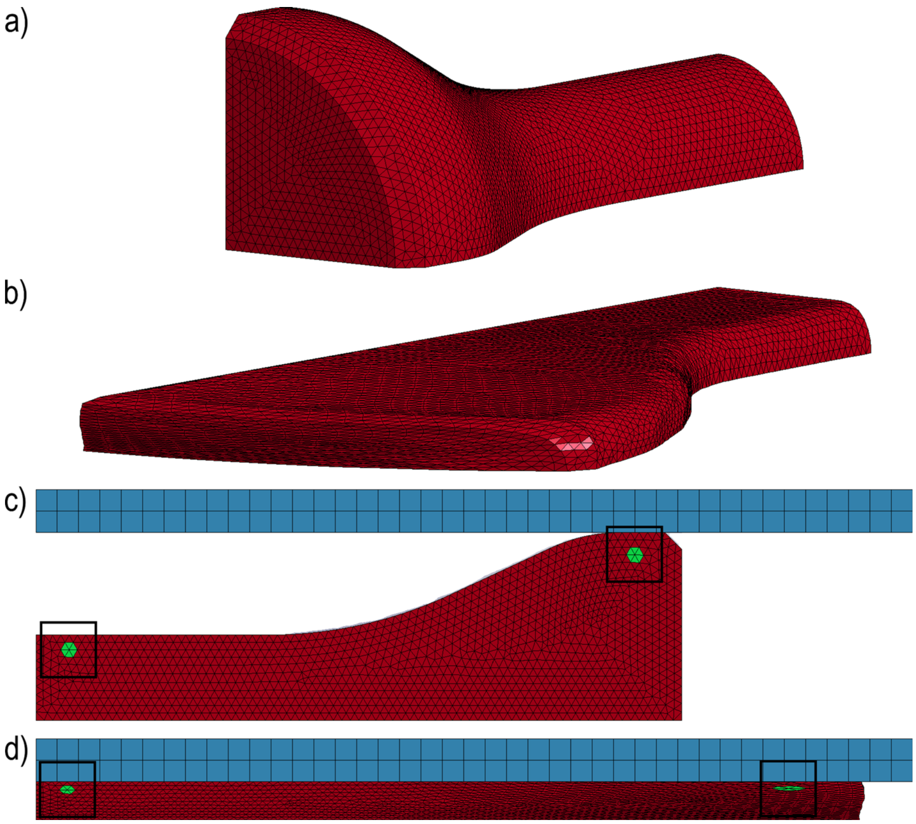

Fig. 14 FE model of a 1/8 bone compression between flat dies; **a** 3D view of initial mesh, **b** 3D view of fully deformed mesh, **c** 2D view of initial mesh and **d** 2D view of fully deformed mesh. The chosen element groups are displayed in green (Color figure online)

to between -3×10^{-3} and 3×10^{-3} mm/s for the cylinder compression and between -5×10^{-3} and 5×10^{-3} mm/s for the bone compression. It is important to make the action range wide enough to allow the RL algorithm to explore the action space and to allow it find actions that lead to rewards even after a series of undesirable decisions, shortening the training time. However, making the action range too wide can lead to long training times and issues with converging on a desirable solution. The stopping condition for each training episode was set to when the value of d exceeded 2.2027 mm (equivalent to 0.8 global true strain) for the cylinder compression and 35.0134 mm (equivalent to 1.6 global true strain) for the bone compression. In addition, the training episode was set to stop if the value of v fell below 0.

Reward function

The reward function is used to direct the behavior of the agent by defining which environment states are desirable and which ones are undesirable. Designing a reward function is a complex task with no universal methodology. In general, a good reward function balances between being concrete enough to lead the agent to a desirable solution and yet abstract enough to allow the agent the opportunity to find optimizations and alternative process routes. This involves precisely defining the desirable environment state. However, this can be difficult as the desirable environment state depends on many factors that can be hard to predict before testing the final RL agent. An example is training a robot arm to carefully move a box. If the reward is measured only in terms of how far the box is moved in distance, the agent can learn to maximize the reward by throwing the box, potentially damaging the contents. The frequency of rewards can also significantly affect the learning process. Sparse rewards or rewards given at the end of an episode can lead to the agent needing to perform many actions before it can determine if a desirable environment state is achieved. This can increase the training time or result in divergencies. Conversely, continuous rewards can lead the agent to desirable solutions faster. In this work, the reward was calculated using σ and was updated in every communication interval. The same reward function was used for both the cylinder and bone compression FE environments. The final reward function used in this work was found through a process of trial and error and is given in Eq. (20):

$$R = \begin{cases} +3, & \text{if } |\Delta\sigma| \le 2\text{ MPa} \\ +1, & \text{if } |\Delta\sigma| > 2\text{ MPa and } |\Delta\sigma| \le 5\text{ MPa} \\ 0, & \text{if } \Delta\sigma < 5\text{ MPa} \\ -30, & \text{if } \Delta\sigma > 5\text{ MPa} \end{cases} \tag{20}$$

where R is the scalar reward, $\Delta\sigma$ is the difference between the measured stress and the desirable goal stress. The goal

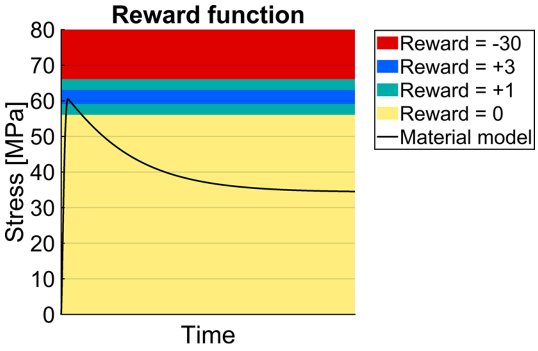

Fig. 15 Plot illustrating the reward function; black line plot: material model flow curve, yellow area: 0 reward, green areas: + 1 reward, blue area: + 3 reward and red area and beyond: – 30 reward (Color figure online)

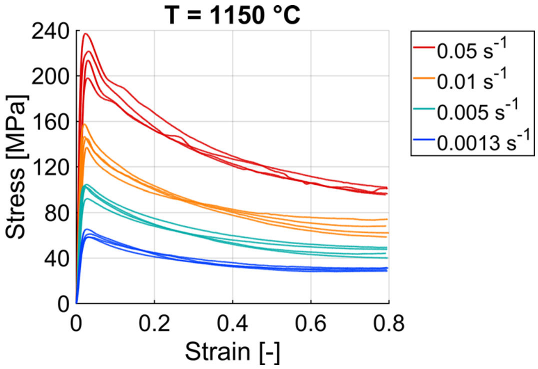

Fig. 16 Experimental flow curves obtained for heat treated TNM-B1 using the constant strain rates 0.0013, 0.005, 0.01 and 0.05 s^{-1} at 1150 °C

stress was set to 61 MPa, as this was the peak stress predicted by the material model at 0.0013 s^{-1} using the median global scale factor of 1.1150.

Figure 15 illustrates the reward function used in this work. The black line plot represents the flow curve predicted by the material model at 0.0013 s^{-1} using a global scale factor of 1.1150. The yellow area represents the stress states where the agent receives 0 reward, the green areas represents + 1 reward, the blue area represents + 3 reward and the red area and beyond represents – 30 reward. In each interval, the reward function was designed such that while σ is below 56 MPa the agent receives 0 reward and while σ is above 66 MPa the agent receives – 30 reward. The yellow area with reward weighting was defined in order to teach the agent that elastic behavior is neutral in terms of desirability. The red area with reward weighting was defined to demonstrate that higher stresses are more undesirable. Furthermore, the negative rewards for higher stresses prevented the agent from taking high velocity shortcuts which led to the flow stress climbing through the red area before settling in the desirable green and blue areas later due to the softening behavior. The widths of the green and blue areas were set as smaller widths led to long training times and divergences. In order to allow the agent to explore different velocity paths without direct restrictions, v was not included in the final reward function. The RL agent was saved if the average cumulative reward over 50 consecutive training episodes exceeded 400 for the cylinder compression and 800 for the bone compression. Finally, a negative reward of – 1000 was given if v fell below 0 and the training episode was stopped. This deterred the agent from exploring velocity paths that led to negative die velocities early during training.

Results and discussion

In this section, the results from the hot compression tests as well as the RL and FE co-simulation for both the cylinder and bone compression environments are presented. The results from the 1/4 2D axisymmetric cylinder FE model and the 1/8 bone FE model were scaled in order to fit the full-sized geometries. Both the cylinder and bone environments were tested by connecting the respective trained ANN controller to the respective FE simulations. In order to evaluate and compare the performances of the ANN controllers and the behaviors of the FE environments, synthetic flow stress surfaces were generated using pseudorandom parameters with the set global scale factors 0.9500, 1.0325, 1.1150, 1.1975 and 1.2800. The same 5 stress surfaces were used as input for both the cylinder and bone compression environments.

Experimental compression tests

Figure 16 displays the flow stress curves resulting from the hot compression tests performed on the heat treated TNM-B1 alloy. The testing conditions used were a constant temperature of 1150 °C and the constant strain rates 0.0013, 0.005, 0.01 and 0.05 s^{-1}. Each strain rate condition was repeated 4 times. The flow curves are characterized by a sharp increase in stress up to a pronounced peak stress, followed by flow softening or stress reduction until steady state. The ranges in measured flow stress were the widest around the peak stress for all tested conditions. They were measured to be around 6.9, 12.3, 20.5 and 39.2 MPa for 0.0013, 0.005, 0.01 and 0.05 s^{-1}, respectively.

Based on the experimental flow curves, assumptions can be made on the range of possible deformation behaviors for

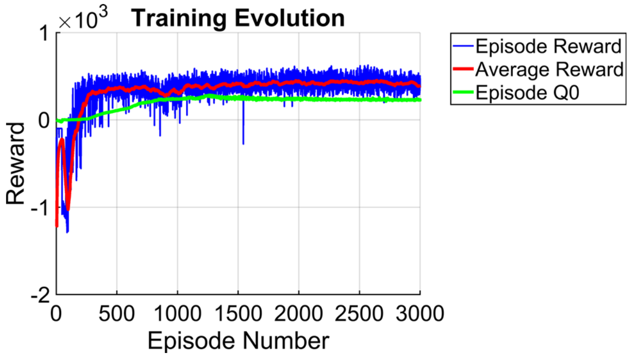

Fig. 17 Training evolution for the cylinder compression environment

TNM-B1. In general, samples displaying a relatively high peak stress also display a relatively high softening path. However, it is possible for the softening rate to be independent of the peak stress. This can be seen especially in the flow curve with the lowest peak stress at 0.01 s^{-1} (orange), as the flow stress ends up as the highest of the group at the end of the stroke. Furthermore, the flow curves display roughnesses and oscillations that increase in scale with increasing strain rate. In particular, the flow curves obtained for 0.05 s^{-1} (red) display a higher scale roughness and higher scale oscillations compared to the flow curves obtained for 0.0013 s^{-1} (blue).

Cylinder compression environment

The RL algorithm "Reinforcement learning fundamentals" section was trained by interacting with an FE model of a 1/4 2D axisymmetric cylinder compression "Finite element model setup" section. The observations were the die force, velocity and displacement and the actions were die velocity adjustments. The action range was set to between -3×10^{-3} and 3×10^{-3} mm/s. The reward function used is presented in "Reward function" section and was calculated based on the difference between the goal stress and the stresses in a chosen element in the workpiece mesh. The goal stress was set to 61 MPa.

Figure 17 displays the evolution from training the RL agent in the cylinder compression environment. The blue line plot represents the reward achieved in each episode, the red line plot represents the average reward achieved over 50 episodes and the green line plot represents the episode Q0, which is the discounted long-term reward at the beginning of each episode predicted by the critic network. Initially, the average rewards were in the range of 0 to − 1000 as the agent explored the minimum and maximum actions in the set action range. Several episodes ended with the die velocity reaching 0, resulting in episode rewards of around − 1000. From around episode 200, the agent began learning that rewards could be collected by adjusting the die velocity such that the element stress was kept in range of the goal stress. From around episode 700 to around episode 1300, the agent explored actions that led to reduced rewards compared to the highest average reward achieved before episode 700. A few episodes around episode 1548 obtained negative episode rewards due to over-accelerating the deformation. From around episode 1550, the training evolution approximately reached steady state, as the agent was not able to collect significantly increased rewards and did not make significant mistakes according to the reward function, compared to previous episodes. Therefore, the training was stopped after 3000 episodes. The final ANN controller used for producing the results was obtained from episode 2395, as this episode achieved a relatively high average reward of around 442.2 over 50 episodes and as the performance of the agent was consistent over a window of around 500 episodes. Each episode took on average around 3 min to complete, giving a total training time of around 9000 min or around 6.3 days to complete 3000 training episodes.

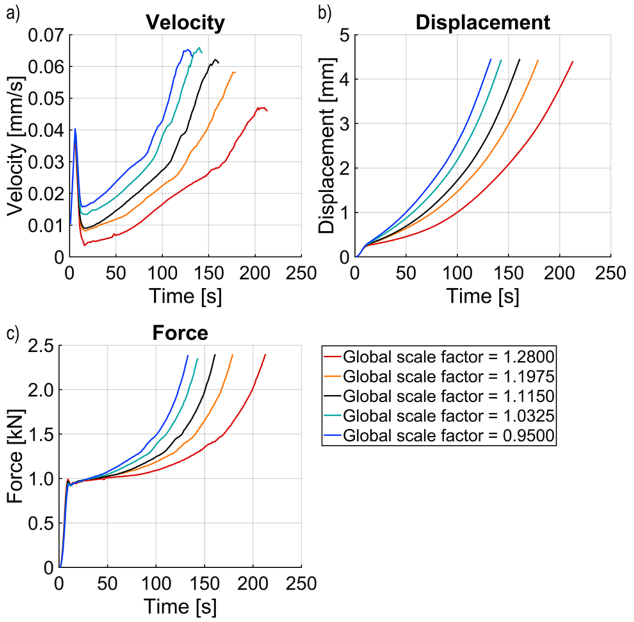

Fig. 18 Observations for the cylinder compression environment; **a** die velocity, **b** die displacement and **c** die force. The input synthetic stress surfaces were generated using the global scale factors 0.9500, 1.0325, 1.1150, 1.1975 and 1.2800

Figure 18 displays the observations (die velocity, displacement and force) over time obtained from testing the trained ANN controller for the cylinder compression environment. The controller predicts velocity profiles that are comparable in shape to the fixed velocity profile displayed in Fig. 5b. The RL agent found an optimization in accelerating the process during elastic deformation. This can be seen in Fig. 18a, as the controller initially applied nearly maximum acceleration until the peak stress was reached for all the tested global scale factors. This was because rewards could be collected faster if the stress reached the range of the goal stress earlier.

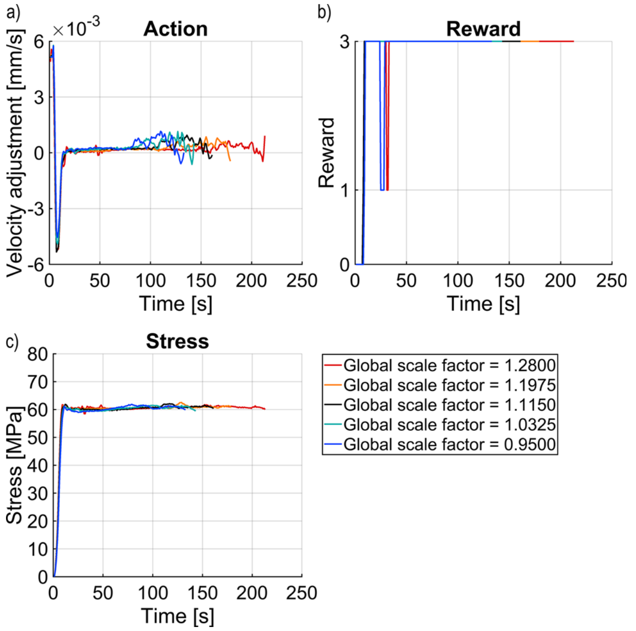

Fig. 19 **a** Action, **b** reward and **c** element stresses for the cylinder compression environment. The input synthetic stress surfaces were generated using the global scale factors 0.9500, 1.0325, 1.1150, 1.1975 and 1.2800

Similar velocity adjustments were made by the controller for all the tested global scale factors during elastic deformation. This was due to the generated synthetic stress surfaces having similar elastic behaviors, producing similar force values before the peak stress was reached. Beyond the peak, the controller adapted the die velocity to the different deformation behaviors of the tested synthetic stress surfaces. The velocity profiles range from the slowest profile corresponding to the global scale factor 1.2800 (red) with a processing time of around 215 s, to the fastest profile corresponding to the global scale factor 0.9500 (blue) with a processing time of around 130 s. This gives a difference of around 85 s in processing time between the highest and the lowest tested global scale factor.

Figure 19 displays the actions (die velocity adjustments), rewards and element stresses over time obtained from testing the trained ANN controller for the cylinder compression environment. As can be seen, the action profiles for all the tested stress surfaces exhibit frequent oscillations with relatively high amplitudes. Particularly from around 80 s until the end of deformation. Furthermore, high acceleration and deceleration was applied initially. For a physical press, repeated and sharp velocity changes can lead to excessive wear and damage to the equipment. In addition, certain press types are not able to quickly adjust the die velocity. However, the behavior of the ANN controller can be improved by tailoring the RL and FE co-simulation to a specific press. For example, the maximum allowable velocity adjustments or the frequency of the adjustments can be limited more appropriately, a more complex FE model that includes the physical aspects and constraints of the press can be used, or the reward function can be expanded to give negative rewards for decisions that lead to equipment being used outside of its tolerance. However, this can involve increased development time, longer training times, requires more knowledge and more complex modeling. The velocity adjustments made by the controller were not wider than the maximum allowable range of $\pm 6 \times 10^{-3}$ mm/s. As can be observed, approximately constant stress during plastic deformation was achieved by the controller for all the tested global scale factors. In every time step, the reward was between 0 and 3 for all the tested global scale factors. The rewards were 1 for the global scale factors 0.9500 (blue) and 1.2800 (red) at around 30 s, as the difference between the stress and the goal stress of 61 MPa became greater than ±2 MPa.

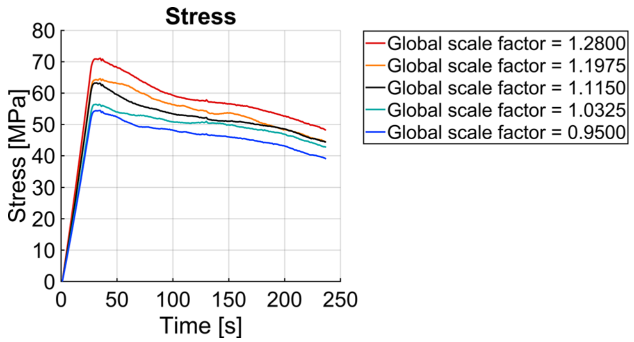

Fig. 20 Element stresses for the cylinder compression environment using the fixed acceleration profile. The input synthetic stress surfaces were generated using the global scale factors 0.9500, 1.0325, 1.1150, 1.1975 and 1.2800

Figure 20 displays the element stresses over time of the cylinder compression environment using the fixed acceleration profile shown in Fig. 5. The roughnesses of the synthetic flow stress surfaces can be observed more distinctly compared to Fig. 19c. As the workpieces were all deformed according to a fixed profile, the processing time was around 235 s for all the tested global scale factors. The maximum difference in flow stress observed using the fixed acceleration profile was around 17 MPa.

Figure 21 displays a histogram of element stresses obtained at 100 s from testing the trained ANN controller for the cylinder compression environment. The results were obtained from 400 tests using synthetic stress surfaces generated with pseudorandom global scale factors varying between 0.9500 and 1.2800. The time of 100 s was used as this was during plastic deformation for all the tested global scale factors. The goal stress of 61 MPa is indicated by the vertical black solid line plot. The stresses approach a normal distribution approximately around 61 MPa. The results show that the stresses were kept inside a range of ±2 MPa of the goal stress for all the tested synthetic stress surfaces.

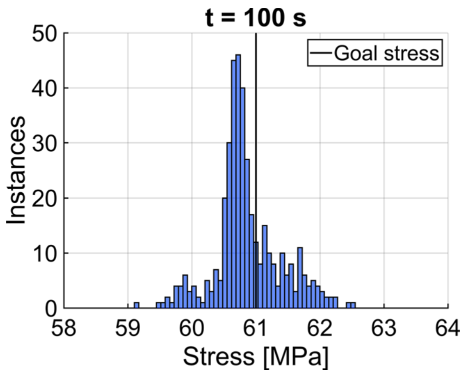

Fig. 21 Element stress histogram at 100 s for the cylinder compression environment. The results were obtained using 400 synthetic stress surfaces generated using pseudorandom global scale factors between 0.9500 and 1.2800

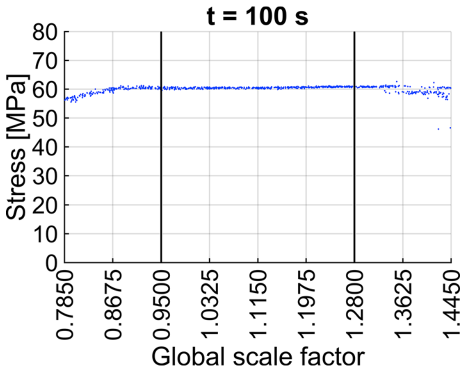

Fig. 22 Element stresses at 100 s for the cylinder compression environment. The results were obtained using 800 synthetic stress surfaces generated using pseudorandom global scale factors between 0.7850 and 1.4450

Figure 22 displays a scatter plot of element stresses obtained at 100 s from testing the trained ANN controller for the cylinder compression environment. The results were obtained from 800 tests using synthetic stress surfaces generated with pseudorandom global scale factors varying between 0.7850 and 1.4450. If the total processing time was below 100 s, as it occasionally was for the lower end range of the scale factors, the element stress from the last time step was used. The limits of the global scale factors used for training are indicated by the vertical black solid line plots. As indicated by the results, the ANN controller performed consistently inside ±2 MPa of the 61 MPa goal stress for the global stress factor interval of 0.9500 to 1.2800 that was used for training. Furthermore, the performance was consistently inside the ±2 MPa range approximately between the global scale factors 0.8700 and 1.3300. Beyond these scale factors in both directions, the element stresses increasingly diverged from the goal stress. This indicates that the developed ANN controller can consistently achieve element stresses near a ±2 MPa range of the 61 MPa goal stress from the global scale factor of around 0.8700 to around 1.3300 for the cylinder compression environment.

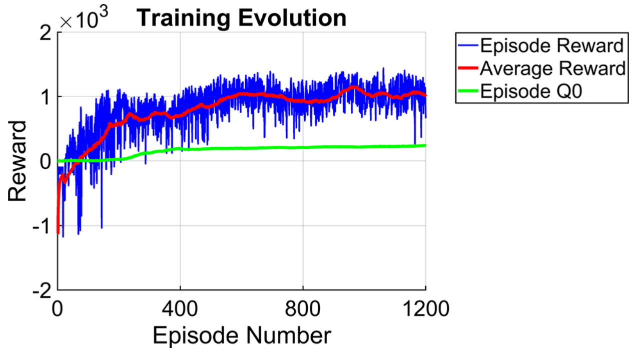

Fig. 23 Training evolution for the bone compression environment

Bone compression environment

The RL algorithm “Reinforcement learning fundamentals” section was trained by interacting with an FE model of a 1/8 3D bone compression “Finite element model setup” section. The observations were the die force, velocity and displacement and the actions were die velocity adjustments. The action range was set to between -5×10^{-3} and 5×10^{-3} mm/s. The reward function used is presented in “Reward function” section and was calculated based on the difference between the goal stress and the stresses in two chosen element groups in the workpiece mesh. The goal stress was set to 61 MPa.

Figure 23 displays the evolution from training the RL agent in the bone compression environment. The blue line plot represents the reward achieved in each episode, the red line plot represents the average reward achieved over 50 episodes and the green line plot represents the episode Q0, which is the discounted long-term reward at the beginning of each episode predicted by the critic network. The episode rewards for the bone compression environment were higher compared to the cylinder compression environment due to the process lasting longer, giving the agent more time steps to collect rewards. Initially, the average rewards were in the range of 0 to − 1000 as the agent explored the minimum and maximum actions in the action range. A few episodes ended with the die velocity reaching 0, as can be seen from the episode rewards of around − 1000 until around episode 180. From this episode, the average rewards steadily increased until around episode 600 where they begin to flatten out. However, the training evolution for the bone compression environment was comparatively less stable than for the cylinder compression environment. The RL agent required more episodes in order to find the actions that led to rewards compared to the cylinder compression environment, which began achieving relatively high rewards after around episode 300

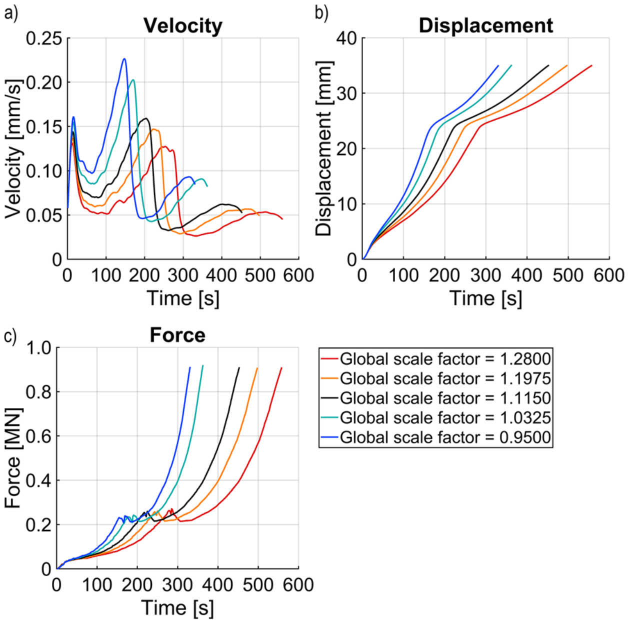

Fig. 24 Observations for the bone compression environment; **a** die velocity, **b** die displacement and **c** die force. The input synthetic stress surfaces were generated using the global scale factors 0.9500, 1.0325, 1.1150, 1.1975 and 1.2800

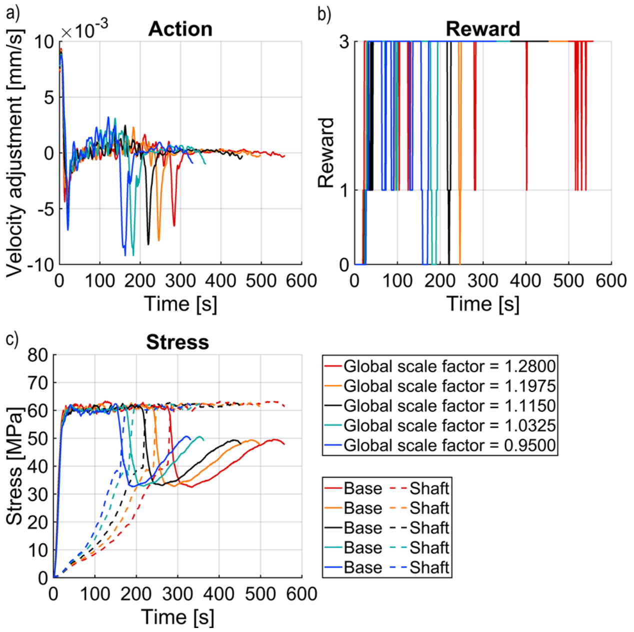

Fig. 25 **a** Action, **b** reward and **c** element stresses for the bone compression environment. The solid line plots are the stresses from the element group in the base part of the bone geometry and the dotted line plots are the stresses from the element group in the shaft. The input synthetic stress surfaces were generated using the global scale factors 0.9500, 1.0325, 1.1150, 1.1975 and 1.2800

(Fig. 17). Furthermore, the RL agent was not able to achieve steady state for a wide window of episodes compared to the cylinder compression environment. This was possibly due to the more complex workpiece geometry, as the agent needed to first accelerate the deformation of the base part of the geometry, decelerate before the die made contact with the shaft and finally accelerate the deformation of the shaft, in order to collect rewards. Thus, making decisions that led to reduced rewards was more likely compared to the cylinder compression environment. Around episode 970, the average rewards peaked with a relatively smaller episode reward range. After this point, the average rewards decreased with a few episodes achieving relatively low rewards. Therefore, the training was stopped after 1200 episodes. The final ANN controller used for producing the results was obtained from episode 968, as this episode achieved a high average reward of around 1147.6 over 50 episodes and as the performance of the agent was consistent over a window of around 250 episodes. Each episode took on average around 10 min to complete, giving a total training time of around 12,000 min or around 8.3 days to complete 1200 training episodes.

Figure 18 displays the observations (die velocity, displacement and force) over time obtained from testing the trained ANN controller for the bone compression environment. As can be seen, the controller predicts velocity profiles that are comparable in shape to the fixed velocity profile displayed in Fig. 7b. As with the cylinder compression environment, the RL agent found the optimization of accelerating the process during the initial elastic deformation. The results show that the controller accelerated the die velocity during the deformation of the base and decelerated before the die made contact with the shaft. The process was subsequently accelerated during plastic deformation of the shaft to a comparatively lesser extent. This behavior was displayed for all the tested global scale factors. The velocity profiles range from the slowest profile corresponding to the global scale factor 1.2800 (red) with a processing time of around 557 s, to the fastest profile corresponding to the global scale factor 0.9500 (blue) with a processing time of around 331 s. This gives a difference of around 226 s in processing time between the highest and the lowest tested global scale factor (Fig. 24).

Figure 25 displays the actions (die velocity adjustments), rewards and element stresses over time obtained from testing the trained ANN controller for the bone compression environment. Both the element stresses from the base (solid line plots) and the shaft (dotted line plots) parts of the bone geometry are shown. As with the cylinder compression environment, the action profiles for all the tested global scale factors exhibit frequent oscillations with high amplitudes. However, the adjustments were increased both in amplitude and frequency compared to the cylinder compression environment. The velocity adjustments were not wider than the maximum allowable range of $\pm 10 \times 10^{-3}$ mm/s. Approximately constant stress during plastic deformation for both the base and the shaft parts of the bone geometry was achieved

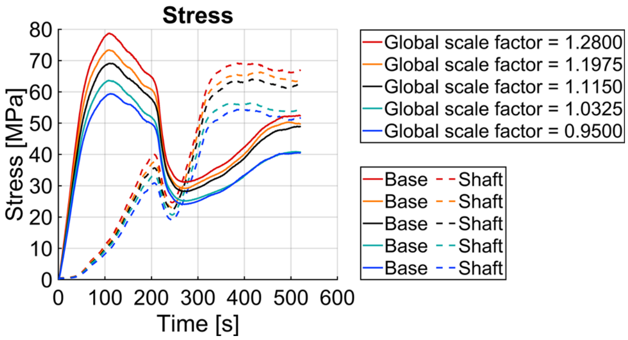

Fig. 26 Element stresses for the bone compression environment using the fixed acceleration profile. The solid line plots are the stresses from the element group in the base part of the bone geometry and the dotted line plots are the stresses from the element group in the shaft. The input synthetic stress surfaces were generated using the global scale factors 0.9500, 1.0325, 1.1150, 1.1975 and 1.2800

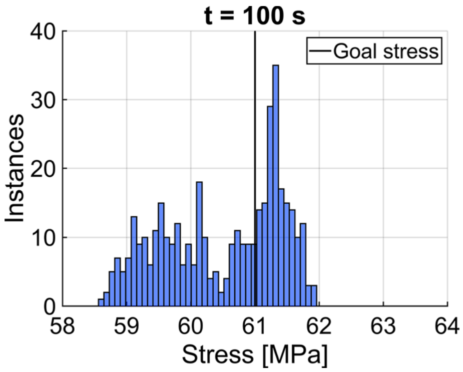

Fig. 27 Element stress histogram at 100 s for the bone compression environment. The results were obtained using 400 synthetic stress surfaces generated using pseudorandom global scale factors between 0.9500 and 1.2800

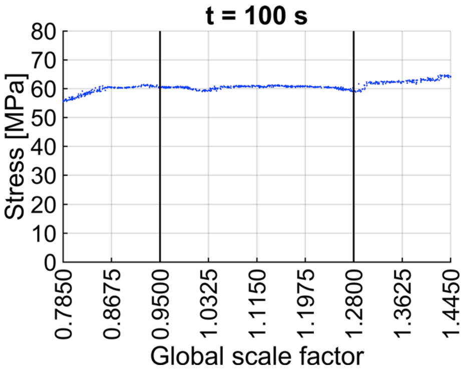

Fig. 28 Element stresses at 100 s for the bone compression environment. The results were obtained using 800 synthetic stress surfaces generated using pseudorandom global scale factors between 0.7850 and 1.4450

for all the tested global scale factors. In every time step, the reward remained between 0 and 3 for all the tested global scale factors. However, time steps where the rewards were 1 and 0 were significantly increased compared to the cylinder compression environment. Furthermore, a relatively higher number of steps achieved 0 reward around the transition between deforming the base to deforming the shaft for the global scale factors 0.9500, 1.0325, 1.1150 and 1.1975, as the stress was 5 MPa or more below the 61 MPa goal stress. Thus, the ANN controller for the bone compression environment was not able to maintain the stress inside the desired ±2 MPa of the 61 MPa goal stress during plastic deformation.

Figure 26 displays the element stresses over time of the bone compression environment using the fixed acceleration profile shown in Fig. 7. The results indicate that using the fixed acceleration profile with the smooth sigmoid transition can avoid significantly increased stress in the shaft compared to the base. However, the dip in stress at around 250 s indicates that this acceleration method results in lost processing time during the transition between deforming the base and deforming the shaft. By comparing the results to the stresses achieved by the ANN controller in Fig. 25c, the RL agent seems to be able to optimize the die velocity in order to avoid this stress dip. The processing time was around 520 s for all the tested global scale factors. The highest observed difference in flow stress using the fixed acceleration profile was around 19 MPa.

Figure 27 displays a histogram of element stresses obtained at 100 s from testing the trained ANN controller for the bone compression environment. The results were obtained from 400 tests using synthetic stress surfaces generated with pseudorandom global scale factors varying between 0.9500 and 1.2800. The goal stress of 61 MPa is indicated by the vertical black solid line plot. The stresses approach a more dispersed distribution with a wider range compared to the cylinder compression environment. The results demonstrate that the stresses were kept inside a range of ±5 MPa of the goal stress for all the tested synthetic stress surfaces.

Figure 28 displays a scatter plot of element stresses obtained at 100 s from testing the trained ANN controller for the bone compression environment. The results were obtained from 800 tests using synthetic stress surfaces generated with pseudorandom global scale factors varying between 0.7850 and 1.4450. The element stress from the last time step was used if the total processing time was below 100 s. The limits of the global scale factors used for training are indicated by the vertical black solid line plots. The results indicate that the stresses achieved by the ANN controller were not as consistent as for the cylinder compression environment. For the global stress factor interval of 0.9500 to 1.2800 that was used for training, a few points were outside of the ±2 MPa range of the 61 MPa goal stress, especially around the global scale factors 1.0300 and 1.2800. From 0.9500 to around 0.8600, the stresses were consistently around the goal stress. Below around 0.8600 and above around 1.2800, the stresses increasingly diverged from the goal stress. This indicates that the developed ANN controller can consistently achieve stresses near a ±5 MPa range of the 61 MPa goal stress using global scale factors from around 0.8600 to around 1.2800 for the bone compression environment.

Conclusions and future work

Conclusions

This study has presented the development of an artificial neural network (ANN) controller using the deep deterministic policy gradient (DDPG) reinforcement learning (RL) algorithm and FE simulations. The aim was to accelerate the hot deformation of the titanium aluminide alloy TNM-B1 (Ti–43.5Al–4Nb–1Mo–0.1B). The ANN controller was trained in two separate FE environments; the compression of a cylinder workpiece between flat dies and the compression of a bone workpiece between flat dies. The RL algorithm interacted with the FE environments by adjusting the die velocity based on the observations die velocity, die displacement and die force. Rewards were given if the stress in a chosen element or group of elements in the workpiece geometry was within a specified range of a goal stress set to 61 MPa. In order to simulate the observed variations in material behavior for the FE environment, a synthetic flow stress surface was generated for each training episode based on the experimental data. In addition, using synthetic data can contribute to reducing overfitting and increasing the real-world applicability of the ANN controller. The experimental data were obtained from performing hot compression tests of the TNM-B1 alloy using the constant strain rates 0.0013, 0.005, 0.01 and 0.05 s^{-1} at a constant temperature of 1150 °C. Surfaces with global scale factors ranging from 0.9500 to 1.2800 were used for training. The cylinder and bone compression FE environments were tested using the final trained ANN controller. For comparison, fixed acceleration profiles controlling the die velocity were tested. The performances of the respective controllers and the behavior of the environments were analyzed and compared based on training evolution, training time, the behavior of the die velocity adjustments, die velocity, die displacement, die force, rewards and stresses in the chosen elements using synthetic flow stress surfaces with the set global scale factors 0.9500, 1.0325, 1.1150, 1.1975 and 1.2800. The same surfaces were used for both the cylinder and bone compression environments. Furthermore, the ability of the ANN controllers to control the process for global scale factors outside the range used for training was investigated. The main conclusions drawn from this investigation are:

- Using the presented RL and FE co-simulation, it was possible to develop an ANN controller that could control the die velocity of the same FE environment used for training. Furthermore, the same RL setup was able to adapt to two different workpiece geometries.
- Given the reward function designed for this work, the RL algorithm was able to find an optimization in accelerating during elastic deformation that was not explicitly programmed.
- For the cylinder compression environment, the ANN controller consistently achieved element stresses near a ±2 MPa range of the goal stress for global scale factors varying from around 0.8700 to around 1.3300. For the bone compression environment, the ANN controller consistently achieved element stresses near a ±5 MPa range of the goal stress for global scale factors varying from around 0.8600 to around 1.2800. This demonstrates that the performance of the trained ANN controller and its ability to make decisions for environment states outside the ones used for training is reduced with increasing workpiece complexity.
- The training evolution for the cylinder compression environment quickly reached high rewards from around episode 200 and reached an approximate steady state from around episode 1550. For the bone compression environment, the training evolution was comparatively less stable and more gradual due to the more complex workpiece geometry. Furthermore, the training time for the bone compression environment (around 8.3 days for 1200 episodes) increased significantly compared to the cylinder compression environment (around 6.3 days for 3000 episodes).

Future work

This work has displayed the potential for using an RL and FE co-simulation setup for developing ANN controllers. However, the controllers were tested in the same simulated environments used for training. The logical next step is to implement a trained ANN controller in a physical press by connecting it to die velocity, force and displacement measurements and allowing it to adjust the die velocity. In order to achieve this, a range of practical problems need to be solved. The performance of the controller strongly depends on the FE environment used for training. Thus, the closer the FE model corresponds to the physical process, the more applicable the final controller will be. Improving the FE model and thereby the controller can involve considerable iterative work, where the controller is tested using the press, the results are checked, the FE model is tuned accordingly, a new controller is trained, repeating the process until desirable results are accomplished. In particular, the FE model has to be tuned such that the observations and actions match the real-world measurements. In this work, the goal of the controller was to achieve a goal stress state. Future work could incorporate more advanced microstructure or damage models into the FE model in order to be able to define more specific and complex goals for the reward function, such as reaching desirable microstructures or damage behaviors in the workpiece. Finally, the RL and FE co-simulation could potentially be used to analyze FE environments and help find alternative process routes and optimizations that are difficult for humans to discover.

Acknowledgements We acknowledge funding received from the German Research Foundation through the Grant BA 4253/4-2 with respect to the work on damage, and through the Grant BA 4253/11-2 for the presented work on property control. In addition, the authors would like to express their gratitude to Thawin Hart-Rawung for advising the model development process.

Funding Open access funding provided by SINTEF. This work has been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation).

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

Achtermann, M., Fürwitt, W., Güther, V., & Nicolai, H. P. (2009). Method for the production of a β-γ-TiAl base alloy, GFE Metalle und Materialien GmbH.

Allam, Z., Becker, E., Baudouin, C., Bigot, R., & Krumpipe, P. (2014). Forging process control: Influence of key parameters variation on product specifications deviations. *Procedia Engineering, 81*, 2524–2529.

Almandoz, G., Ugalde, G., Poza, J., & Escalada, A. J. (2012). Matlab-simulink coupling to finite element software for design and analysis of electrical machines. *A Fundamental Tool for Scientific Computing and Engineering Applications* (pp. 161–184).

Anderson, J. A. (1995). *An introduction to neural networks*. MIT Press.

Bambach, M., & Imran, M. (2019). Extended Gurson-Tvergaard-Needleman model for damage modeling and control in hot forming. *CIRP Annals, 68*, 249–252.

Bambach, M., Sizova, I., Bolz, S., & Weiss, S. (2016). Devising strain hardening models using Kocks-Mecking plots—a comparison of model development for titanium aluminides and case hardening steel. *Metals, 6*, 204.

Beaulieu-Jones, B. K., Wu, Z. S., Williams, C., Lee, R., Bhavnani, S. P., Byrd, J. B., & Greene, C. S. (2019). Privacy-preserving generative deep neural networks support clinical data sharing. *Circulation: Cardiovascular Quality and Outcomes, 12*, e005122.

Beddoes, J., Zhao, L., Immarigeon, J. P., & Wallace, W. (1994). The isothermal compression response of a near γ-TiAl + W intermetallic. *Materials Science and Engineering: A, 183*, 211–222.

Belytschko, T., & Bindeman, L. P. (1993). Assumed strain stabilization of the eight node hexahedral element. *Computer Methods in Applied Mechanics and Engineering, 105*, 225–260.

Bewlay, B. P., Nag, S., Suzuki, A., & Weimer, M. J. (2016). TiAl alloys in commercial aircraft engines. *Materials at High Temperatures, 33*, 549–559.

Brotzu, A., Felli, F., Marra, F., Pilone, D., & Pulci, G. (2018). Mechanical properties of a TiAl-based alloy at room and high temperatures. *Materials Science and Technology, 34*, 1847–1853.

Brunton, S. L., & Kutz, J. N. (2019). *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press.

Camacho, E. F., & Alba, C. B. (2013). *Model predictive control*. Springer.

Choi, E., Biswal, S., Malin, B., Duke, J., Stewart, W. F., Sun, J. (2017). Generating multi-label discrete patient records using generative adversarial networks. In *Machine learning for healthcare conference* (pp. 286–305).

Cingara, A., & McQueen, H. J. (1992). New formula for calculating flow curves from high temperature constitutive data for 300 austenitic steels. *The Journal of Materials Processing Technology, 36*, 31–42.

Clemens, H., & Kestler, H. (2000). Processing and applications of intermetallic γ-TiAl-based alloy. *Advanced Engineering Materials, 2*, 551–570.

Clemens, H., & Mayer, S. (2013). Design, processing, microstructure, properties, and applications of advanced intermetallic TiAl alloys. *Advanced Engineering Materials, 15*, 191–215.

Clemens, H., & Smarsly, W. (2011). Light-weight intermetallic titanium aluminides—status of research and development. *Advanced Materials Research, 278*, 551–556.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems, 2*, 303–314.

Doshi-Velez, F., & Kim, B. (2017). A roadmap for a rigorous science of interpretability, arXiv:1702.08608.

Eisentraut, M., Bolz, S., Sizova, I., Bambach, M., & Weiss, S. (2019). Development of a heat treatment strategy for the γ-TiAl based alloy TNM-B1 to increase the hot workability. *SN Applied Sciences, 1*, 1516.

Ernst, D., Glavic, M., Capitanescu, F., & Wehenkel, L. (2009). Reinforcement learning versus model predictive control: A comparison on a power system problem. *IEEE Transactions on Systems, Man, and Cybernetics Part B (Cybernetics), 39*, 517–529.

Esteban, C., Hyland, S. L., & Rätsch, G. (2017). Real-valued (Medical) time series generation with recurrent conditional GANs. arXiv:1706.02633.

Garcia, N., & Stoll, E. (1984). Monte Carlo calculation for electromagnetic-wave scattering from random rough surfaces. *Physical Review Letters, 52*, 1798–1801.

Ghanta, S., Subramanian, S., Sundararaman, S., Khermosh, L., Sridhar, V., Arteaga, D., Luo, Q., Das, D., & Talagala, N. (2018). Interpretability and reproducability in production machine learning applications. In *17th IEEE International Conference on Machine Learning and Applications (ICMLA)* (pp. 658–664).

Görges, D. (2017). Relations between model predictive control and reinforcement learning. *IFAC, 50*, 4920–4928.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks, 2*, 359–366.

Hutter, F., Kotthoff, L., & Vanschoren, J. (2019). *Automated machine learning*. Springer.

Islam, R., Henderson, P., Gomrokchi, M., & Precup, D. (2017). Reproducibility of benchmarked deep reinforcement learning tasks for continuous control, arXiv:1708.04133.

Jordon, J., Yoon, J., & Schaar M. V. D. (2018). Measuring the quality of synthetic data for use in competitions, arXiv:1806.11345.

Kaloskampis, I., Pugh, D., Joshi, C., & Nolan, L. (2019). *Synthetic data for public good*. Data science for public good.

Kim, Y. W., & Boyer, R. R. (1991). Microstructure/property relationships in titanium aluminides and alloys. *Minerals, Metals, and Materials Society* (p. 237).

Kormushev, P., Calinon, S., & Caldwell, D. G. (2013). Reinforcement learning in robotics: Applications and real-world challenges. *Robotics, 2*, 122–148.

Kumar, A., Grandhi, R. V., Chaudhary, A., & Irwin, D. (1992). *Process Modelling and Control in Metal Forming, 1992 American Control Conference, Chicago* (pp. 817–821).

Lemons, D. S., & Gythiel, A. (1997). Paul Langevin's 1908 paper "On the Theory of Brownian Motion" ["Sur la théorie du mouvement brownien", C. R. Acad. Sci. (Paris) 146, 530-533 (1908)]. *American Journal of Physics, 65*, 1079–1081.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N. M. O., Erez, T., Tassa, Y., Silver, D., & Wierstra, D. (2016). CoRR: Continuous control with deep reinforcement learning.

Mikail, R., Husain, I., & Islam, M. (2013). Finite element based analytical model for controller development of switched reluctance machines. In *2013 IEEE Energy Conversion Congress and Exposition* (pp. 920–925).

Millett, J. C. F., Brooks, J. W., & Jones, I. P. (1993). Large-scale deformation of a single-phase TiAl alloy at elevated temperatures. *Materials and Design, 14*, 61–63.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A., Veness, J., Bellemare, M., Graves, A., Riedmiller, M., Fidjeland, A., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature, 518*, 529–533.

Mueller, T., Kusne, A., & Ramprasad, R. (2016). Machine learning in materials science: Recent progress and emerging applications. *Reviews in Computational Chemistry, 29*, 186–273.

Olorisade, B. K., Brereton, P., & Andras, P. (2017). Reproducibility in machine learning-based studies: An example of text mining. *Journal of Biomedical Informatics, 73*, 1–13.

Orszulik, R. R., & Gabbert, U. (2016). An interface between Abaqus and Simulink for high-fidelity simulations of smart structures. *IEEE/ASME Transactions on Mechatronics, 21*, 879–887.

Pearson, K. (1901). LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 2*, 559–572.

Sallab, A., Abdou, M., Perot, E., & Yogamani, S. (2017). Deep reinforcement learning framework for autonomous driving. *Electronic Imaging, 2017*, 70–76.

Shawi, R. E., Maher, M., & Sakr, S. (2019). Automated machine learning: State-of-the-art and open challenges, arXiv:1906.02287.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., Lillicrap, T., Simonyan, K., & Hassabis, D. (2018). A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science, 362*, 1140–1144.

Silver, D., Lever, G., Heess, N. M. O., Degris, T., Wierstra, D., & Riedmiller, M. A. (2014). Deterministic policy gradient algorithms. In *31st International Conference on Machine Learning, ICML* (pp. 387–395).

Stendal, J. A., Bambach, M., Eisentraut, M., Sizova, I., & Weiss, S. (2019). Applying machine learning to the phenomenological flow stress modeling of TNM-B1. *Metals, 9*, 220.

Stendal, J. A., Eisentraut, M., Imran, M., Sizova, I., Bolz, S., Weiß, S., & Bambach, M. (2021). Accelerated hot deformation and heat treatment of the TiAl alloy TNM-B1 for enhanced hot workability and controlled damage. *Journal of Materials Processing Technology, 291*, 116999.

Strano, M., Jirathearanat, S., & Altan, T. (2001). Adaptive FEM simulation for tube hydroforming: A geometry-based approach for wrinkle detection. *CIRP Annals, 50*, 185–190.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT Press.

Tvergaard, V., & Needleman, A. (1984). Analysis of the cup-cone fracture in a round tensile bar. *Acta Metallurgica, 32*, 157–169.

Uhlenbeck, G. E., & Ornstein, L. S. (1930). On the theory of the Brownian motion. *Physical Review, 36*, 823–841.

Xie, L., Lin, K., Wang, S., & Wang, F., Zhou, J. (2018). Differentially private generative adversarial network. arXiv:1802.06739.

Zhang, C., Vinyals, O., Munos, R., & Bengio S. (2018). A study on overfitting in deep reinforcement learning, arXiv:1804.06893.

Zhang, Z., Dequidt, J., Kruszewski, A., Largilliere, F., & Duriez, C. (2016). Kinematic modeling and observer based control of soft robot using real-time Finite Element Method. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 5509–5514).

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.