Research Article Vol. 30, No. 12 / 6 Jun 2022 / Optics Express 20963

Optics EXPRESS

Motion control for laser machining via reinforcement learning

YUNHUI XIE, MATTHEW PRAEGER, JAMES A. GRANT-JACOB, ROBERT W. EASON, AND BEN MILLS[*]

Optoelectronics Research Centre, University of Southampton, Southampton, SO17 1BJ, UK
[*]*b.mills@soton.ac.uk*

Abstract: Laser processing techniques such as laser machining, marking, cutting, welding, polishing and sintering have become important tools in modern manufacturing. A key step in these processes is to take the intended design and convert it into coordinates or toolpaths that are useable by the motion control hardware and result in efficient processing with a sufficiently high quality of finish. Toolpath design can require considerable amounts of skilled manual labor even when assisted by proprietary software. In addition, blind execution of predetermined toolpaths is unforgiving, in the sense that there is no compensation for machining errors that may compromise the quality of the final product. In this work, a novel laser machining approach is demonstrated, utilizing reinforcement learning (RL) to control and supervise the laser machining process. This autonomous RL-controlled system can laser machine arbitrary pre-defined patterns whilst simultaneously detecting and compensating for incorrectly executed actions, in real time.

1. Introduction

A diverse range of applications exist for laser materials processing [1–7]. In general, laser energy is directed onto a target material, typically with energy concentration via focusing optics, and causes a chemical or physical change in that target material [8–10]. The intended effects of the laser energy are various; for example, in laser machining, the aim may be to remove part of the target material via ablation [11–13], whilst in laser welding the aim is to connect two parts by causing a phase change and creating a melt-pool which flows and solidifies to make the join [14–16]. Many applications of laser processing require positioning of the laser energy on the target material at a level of precision that cannot be achieved manually, and therefore computerized control hardware is typically needed [17,18]. Beam positioning can be achieved in many ways, for example: via motorized stages that move the workpiece [19] or the beam delivery optics [20]; via galvanometer scanning mirrors that steer the laser beam (typically used in conjunction with an f-theta lens) [21]; via spatial light modulation, e.g., with a digital micromirror device (DMD) [22]. In practice, a solution may involve all three of these strategies in combination [23].

Regardless of the physics of the light-matter interaction involved and the specifics of the laser beam control hardware, a common challenge in laser processing is the selection (via an algorithm) of an appropriate set of laser positioning coordinates, or a trajectory that should be followed, to achieve the intended result for the application. Perhaps the simplest conventional example of such an algorithm is a raster scan in which the intended laser machining pattern is digitized into a binary image, with a pre-defined relationship between the pixel size and the physical scale of laser markings on the workpiece. The laser beam is scanned across each row of pixels one-by-one with the laser turning on and off in accordance with the binary value of the pixel. Examples of the raster scan method for laser machining target patterns are illustrated in Fig. 1(a)) and b). The target shape (the yellow shaded region, which can be parametrically defined, as in a

#454793 https://doi.org/10.1364/OE.454793
Journal © 2022 Received 31 Jan 2022; revised 7 Apr 2022; accepted 17 Apr 2022; published 26 May 2022

vector image format) is digitized into pixels, as indicated by the grid; with the pixel outline colors indicating the laser action (gray for off and black for on). Note that, in both Fig. 1(a)) and b) the physical extent of laser machining for each pulse (which can be inferred from the green curve that shows the final machining outline) is larger than the pixel size chosen when the target shape was digitized. In fact, in Fig. 1(a)), the pixel width is set equal to $\sqrt{2}$ times the radius of the laser pulse markings (this is the condition which, for a circular beam, ensures complete machining of each pixel with the minimum overlap into neighboring pixels). In Fig. 1(b)) the target shape is digitized with a smaller pixel size (equal to the radius of laser machining); this causes more overlap between neighboring laser machining pulses (potentially machining more deeply) but does allow the perimeter of the target shape to be defined with greater precision. This simple but straightforward control method has been widely adopted in the field of laser processing [24–26].

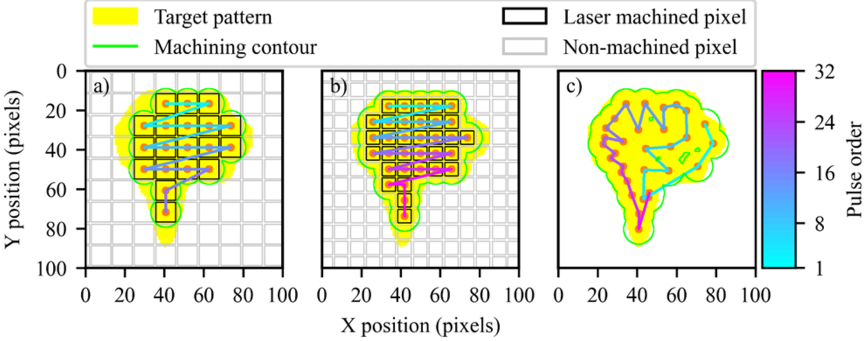

Fig. 1. a) Illustration of the raster scan approach for laser machining of a randomly generated target pattern. 19 incident pulses successfully machine 82.04% of the target pattern, but inadvertently machine 2.16% of the target pattern area from the surrounding material. In a) the target pattern is digitized (as indicated by the grid) with pixels that are $\sqrt{2}$ times larger than the radius of laser pulse markings. The grid color indicates pixels that should be machined (black) and those that should not (gray). In b) the target pattern is digitized with pixels that are equal in width to the laser machining radius. In this case 33 laser pulses successfully machine 89.32% of the target pattern, and inadvertently machine 1.51% of the target pattern area from the surrounding material. c) Using the RL approach presented in this paper on the same target pattern, 30 pulses successfully machine 97.21% the target area, with 5.51% of the area of the target pattern being inadvertently machined from the surrounding material.

Reinforcement learning (RL) has gained popularity in recent years, as it provides a unique capability for observing and responding to systems of great complexity. In photonics, previous studies have utilized RL for a wide range of applications, such as coherent beam combining [27], laser alignment [28] and optical tweezers control [29]. Similarly, laser materials processing control systems, actuated by RL, have been devised for laser welding [30]. In Ref. [30], dimension reduction is performed based on camera observations of the laser welding process via an auto-encoder, and a deep neural network is employed to predict state transitions on highly abstract representations of those camera observations, enabling a RL program to learn laser power control from real-world laser welding data. This proposed controlling paradigm is very similar to the model-based RL algorithm WorldModel [31]. In Ref. [32], a similar RL paradigm to [30] is adopted for controlling laser power in laser welding. This work is particularly interesting because multiple sensors (i.e., photodiodes working at different wavelengths and an acoustics sensor) are employed concurrently as sources of information for the RL program to learn from.

In this current work, we report a novel approach that uses RL for automatic toolpath design and allows real-time feedback via a camera image of the workpiece. The RL agent is trained in a virtual environment that approximates the real-world laser machining experiment, thus minimizing reliance on real-world data collection. We then demonstrate that, once trained, the RL agent can be applied in the physical environment and that real-world camera observations of the workpiece can be processed to achieve the same visual appearance as the virtual environment (making them intelligible to the RL agent). Whilst Fig. 1(a)) and b) illustrated the raster scan method, Fig. 1(c)) shows the toolpath generated by the RL approach for the same target pattern. The RL approach efficiently fills the interior of the target pattern with relatively few laser pulses and more closely follows the exterior contour of the target pattern. The performance of the RL method is further explored in the results section where we benchmark the RL approach by comparing it with a state-of-the-art conventional optimization algorithm. In section 3.4, we demonstrate that the RL approach is not only capable of designing toolpaths for pre-defined target patterns, but can also monitor the machining process and update its toolpath design *in real time* to correct machining errors (for example, those caused by unexpected stage vibrations which we simulate by deliberately mis-positioning selected laser pulses).

2. Method

2.1. Experimental apparatus and details

Laser machining experiments were carried out using a Light Conversion Pharos-SP-1mJ laser system, producing 190 fs pulses with a center wavelength of 1025 nm and pulse energy of up to 1 mJ. For all experiments, the target material was a 1 mm thick fused silica microscope slide that was machined using a fluence of ~4.6 J/cm^2. The laser beam was focused onto the target material (workpiece) using a Nikon 20x microscope objective with an N.A. of 0.4. As shown schematically in Fig. 2(a)), the laser beam was reflected from a beam splitter, which allowed simultaneous imaging of the target material using a CMOS camera (Thorlabs DCC1645C). The workpiece was mounted on an XYZ motorized translation stage, with the Z axis determining the laser focus, (the Z position was manually adjusted and set constant for these experiments). Machining of the workpiece was conducted one laser pulse at a time, with each individual pulse producing a visible modification to an approximately circular region of the workpiece with a diameter of ~18 µm.

RL laser machining experiments follow a sequence of events (presented in Fig. 2(a)) that are repeated until the RL agent detects an exit condition (such as when a threshold percentage of the target pattern has been correctly machined or when the maximum allowed number of pulses has been reached). The sequence is as follows: The motorized XY stages are moved to the imaging position, and a camera observation of the workpiece is made (this is the "before" observation). The RL agent is then shown the target pattern and decides the XY coordinates at which the next laser machining pulse should be applied. This action is executed by moving the XY stages to the selected position and triggering a single laser pulse. The XY stages then return to the imaging position and make the "after" camera observation. Image processing techniques, namely image subtraction, image denoising and applying a Hough transform, are applied to the "before" and "after" camera images to identify regions of the workpiece that have been machined. This information is then used to update the target image (removing parts that have been correctly machined). During training, the virtual environment also calculates a reward signal that is returned to the RL agent. The sequence continues, with the "after" observation for pulse N becoming the "before" observation for pulse N+1. This loop is halted when the RL agent determines that the fidelity of the laser-machined workpiece (with reference to the target pattern) cannot be further improved by applying additional laser pulses.

An artificial example of the input to the RL is shown in Fig. 2(b)). The target pattern was initially a circle but has been updated following each laser pulse in a sequence that marks a diagonal line across the circle. Note that there are just two allowed values for pixels in the target image; these denote the region that is still to be machined (the "remaining target" in black), and

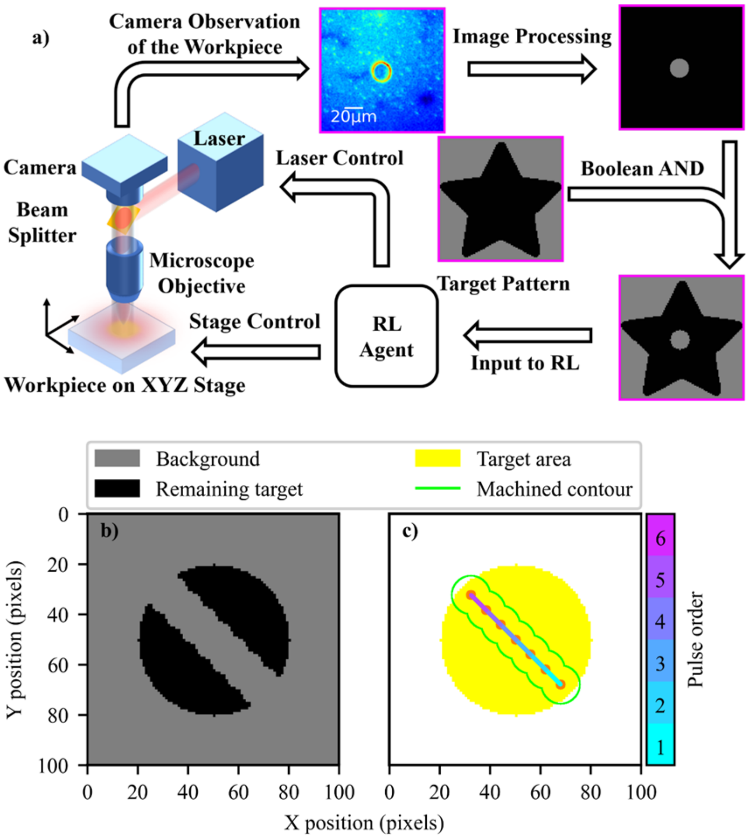

Fig. 2. a) Schematic of the experiment setup, where a laser pulse is incident at the center of a workpiece. The target pattern here is a five-pointed star. b) An artificial example of an input to the RL agent, where the target pattern is a circle that has already been partially machined by a diagonal line of laser pulses. c) A visual representation of the laser machining trajectory that has occurred for the target pattern shown in b). In c), the initial target pattern is shaded in yellow, the centers of pulses are marked with red dots and a color-coded line links the dots to show the order of machining (the trajectory). A green curve is also included which shows the simulated extent of laser machining.

the region that should not be machined, or has already been machined (the "background" in gray). Whilst Fig. 2(b)) represents the state of a laser machining experiment at one instant in time, Fig. 2(c)) includes additional information and represents the laser machining trajectory for the whole experiment (in a visualization style intended for humans). This visualization style is used in figures throughout this manuscript. The yellow shaded region shows the initial target pattern (a circle in this case), red dots mark the coordinates where laser pulses have been applied that are joined by a color-coded line that indicates the order of laser machining and a green contour shows the final extent of laser machining.

Comparing Fig. 2(b)) and Fig. 2(c)), it is evident that during the course of laser machining, pixels in the diagonal line of the target image shown in Fig. 2(b)) have been reassigned from their initial status as "remaining target" to "background" status. Correctly identifying areas of the workpiece that have already been laser-machined (to update the target image) is one of the most crucial parts of the experiment. The image processing procedure for identifying a laser-machined region is as follows: 1) Image subtraction of the microscope camera observations of the workpiece before and after the laser pulse is used to find parts of the image that have changed. 2) Image denoising. 3) The Hough transform is used to identify the coordinates and size of circular laser-machined features. These image manipulation methods are all of common use, are available on many platforms and are not compute-intensive; for a low-end desktop configured with a 4th generation Intel i5 CPU, it takes approximately 0.15 seconds to process per image, without any code optimization. By repeating this image processing procedure for each laser pulse, the relative positions of all laser-machined features on the workpiece can be known. From this process, pixel values in the target pattern can be updated based on the actual laser-machined features on the workpiece (rather than on the coordinates specified by the RL agent). This means that, in the event of laser pulse inadvertently deviating from its expected position (perhaps due to stage vibrations), the RL agent has the chance to adjust its toolpath design, in real time, in order to compensate for this error.

2.2. Proximal Policy Optimization (PPO)

RL algorithms are currently an active topic of research, with rapid development in the performance level they can achieve on standardized test environments [33], the efficiency with which they can be trained, and the scope of applications to which they can be applied [34]. In this work we do not attempt to develop new RL algorithms; rather, we aim to implement a near state-of-the-art algorithm to the task of path planning, in the specific use-case of laser machining. Although seemingly the specific implementation of an RL algorithm should have little impact on its performance, in actuality, nuances in implementation of the same RL pseudocode can result in significant variations in their respective performance [35]. Therefore, in the spirit of reproducibility and accessibility, we based our work on an open-source platform Stable Baselines [36]. At the beginning of this work, we took the decision that it was desirable for the RL algorithm to control the laser machining operation though continuously variable actions (rather than discrete values). This type of problem ("continuous control") is typically more difficult for an RL algorithm to learn; however, it was decided that this offered the potential for more precise control, for example, if the challenge was later extended to include parameters such as laser power. Of the RL algorithms included in Stable Baselines, the following are currently suggested as being suitable for continuous control problems: A2C [37], DDPG [38], GAIL [39], PPO [40], SAC [41], TD3 [42], TRPO [43]. The OpenAI Gym CarRacing-v0 environment presents a continuous control task that is to be learned from image type observations, we consider this to be comparable in complexity to our laser machining task. The current leaderboard (https://github.com/openai/gym/wiki/Leaderboard, as of 6th April 2022) for high scores achieved on this environment contains several algorithms based on Proximal Policy Optimization (PPO), including the current highest score [44]. For this reason, here we chose to adopt the PPO algorithm which, in its base form, optimizes a Gaussian distributed policy (see supplementary Table S1 for hyperparameters). Additionally, self-imitation learning [45,46] is incorporated into this implementation of the PPO algorithm to improve exploration (see supplementary Table S2 for hyperparameters). In order to minimize the necessity to fine-tune hyperparameters, we followed the common practice of normalizing the action space (i.e., the upper and lower bounds for actions were always between -1 and 1) and enabled reward and input normalization (i.e., rewards and inputs were normalized by their running mean and standard deviation respectively). This allowed us to adopt hyperparameter values that had previously been determined to be near optimal for similarly normalized environments from RL-Baselines-Zoo library [47]. It should therefore be noted that the RL results here represent a feasibility demonstration for the application of laser machining, rather than a fully optimized solution, and that further small improvements in performance could be expected if full parameter optimization was undertaken.

2.3. Training in the virtual environment

A virtual training environment (commonly known as a “gym”) is used in this work, which allows the RL agent to learn a strategy of selecting XY positions without having to acquire actual experience in the physical experiment. This “gym” approach has recently become popular because: 1) It is sometimes prohibitively expensive (computationally or/and financially) to utilize real-world data for training. 2) Explorative actions of a freshly initiated RL agent can be dangerous to execute on experimental apparatus. We developed a virtual environment (based on the OpenAI Gym framework [48]) which approximates the experimental setup, and we refer to this as “the virtual environment” in the remainder of this work.

The virtual environment gives the RL agent the opportunity to practice toolpath generation rapidly and repeatedly for chosen or automatically created target patterns. The virtual environment operates in a similar manner to the experimental apparatus, with the RL agent working through a cycle of repeating events, namely, observing a virtual workpiece, selecting an XY position based on the observation of the virtual workpiece, moving a virtual stage to the selected XY position and applying a virtual laser pulse. The differences between the virtual and the physical environment are that 1) image processing is not needed since inputs to the RL agent can be simulated directly, and 2) a figure of merit (commonly referred to as the “reward”) is granted to the RL agent following the completion of each cycle. This reward is a numerical value that increases in proportion to the area of the “remaining target” region that was correctly laser machined in the current cycle and decreases for any pixels that were incorrectly machined from the “background” region. This scheme of reward-granting incentivizes the RL agent to machine the “remaining target” region rather than the “background” region, and hence helps the RL agent to develop a strategy of selecting appropriate XY positions from the “remaining target” region of a given target pattern.

2.4. Covariance Matrix Adaptation Evolution Strategy (CMA-ES)

Throughout this work, to benchmark the capability of our RL agent, we compare its performance to the well-studied Covariance Matrix Adaptation Evolution Strategy (CMA-ES) algorithm [49,50]. Unlike the RL agent, which selects XY positions one-by-one from observations of the workpiece and adjusts its strategy for selecting XY positions based on the received reward; the CMA-ES iteratively searches for an entire set of XY positions that maximize received reward for the given target shape (without any subsequent reference to the workpiece). Whilst the two approaches differ in terms of their mechanics for toolpath generation, the sets of XY positions that they find are comparable, and hence the relative performance of the RL approach, can be quantitatively evaluated.

It must be clearly stated that we expect the CMA-ES algorithm to find a coordinate set that is very close to, if not actually, the optimal solution for any given target shape. It is therefore likely impossible for the RL agent to exceed the performance achieved by CMA-ES, in fact, we fully anticipate that the RL agent will fall short of this benchmark. Many of the arbitrarily generated target shapes, used during training and testing, contain features that are smaller than the focused laser spot, and are therefore impossible to machine without also removing some of the surrounding material. This makes it very difficult to calculate analytically the theoretical maximum reward achievable for a given target shape; the CMA-ES results therefore offer a convenient means to estimate the likely upper bound of performance. Even with slightly lower absolute performance, the RL approach offers two key advantages that could make it more practical than CMA-ES for laser machining toolpath generation namely; 1) Once trained, an RL agent can potentially generate coordinate sets for any arbitrary target shape quickly and with very low computational requirements (CMA-ES must perform lengthy optimization calculations each time it encounters a new target pattern); 2) The implementation of the RL agent (one laser pulse at a time) and its capability to make observations of the workpiece during machining give the RL approach the potential to react to and therefore correct machining errors in real time during the machining process.

3. Results and discussion

3.1. Static target pattern

Our goal in this work is to produce a generalized RL agent capable of selecting XY machining positions that maximize reward for any given target pattern. However, in this first section we tackle a simpler problem and demonstrate that our RL agent is able to learn such a strategy, via a large number of interactions with a virtual environment, where the initial target shape is kept constant.

Figure 3(a)) presents, for a virtual environment where the initial target shape is always the same five-pointed star, the moving-average of reward per training episode as training progresses (referred to as a “training curve”). Initially rewards increase quite rapidly, indicating that the RL agent is learning and improving its strategy. Eventually however the rewards plateau, indicating that the best performance has been achieved (given the RL agent’s current internal hyperparameters, such as discount factor). From this figure, it can be observed that the training curve is not very smooth and has low reward outliers occurring during training. This is a strong indication that fine-tuning of the hyperparameters could yield improvements in learning and perhaps in eventual RL agent performance. However, due to our limited computational resources, and our aim for proof-of-principle rather than a fully optimized solution, further hyperparameter tuning was not performed in this work. As expected, the moving-average reward for the RL agent stabilizes at a value that is slightly lower than the optimal value achieved by CMA-ES. A likely reason for this shortfall is that the PPO algorithm optimizes a Gaussian distributed policy, which introduces a degree of random variation into the XY positions that it selects during training. Trajectories from the CMA-ES and PPO RL methods (from the end of the training period) are compared in Fig. 3(b)) left and right, respectively (see Table 1 row (a) for a quantitative comparison). The fact that the RL agent is able to design a toolpath that achieves comparable machining performance, and with fewer laser pulses than the CMA-ES method clearly demonstrates that, at least for the fixed star-shaped target pattern, the PPO algorithm was able to learn the laser machining task in our virtual environment.

We subsequently applied our RL agent to the physical environment in which the RL agent controls both the XY translation stages and the laser, the result of which is presented in Fig. 3(c)). In this laser machining experiment, a total number of 41 laser pulses were applied to the physical workpiece. It is observable that subtle differences exist in the laser pulse positions selected by the RL agent in the simulation (Fig. 3(b)) right) and the physical experiment (Fig. 3(c))), with an additional laser pulse also being applied in the physical experiment. The origins of these slight differences are twofold: 1) Image processing errors can arise from varying illumination

Optics EXPRESS

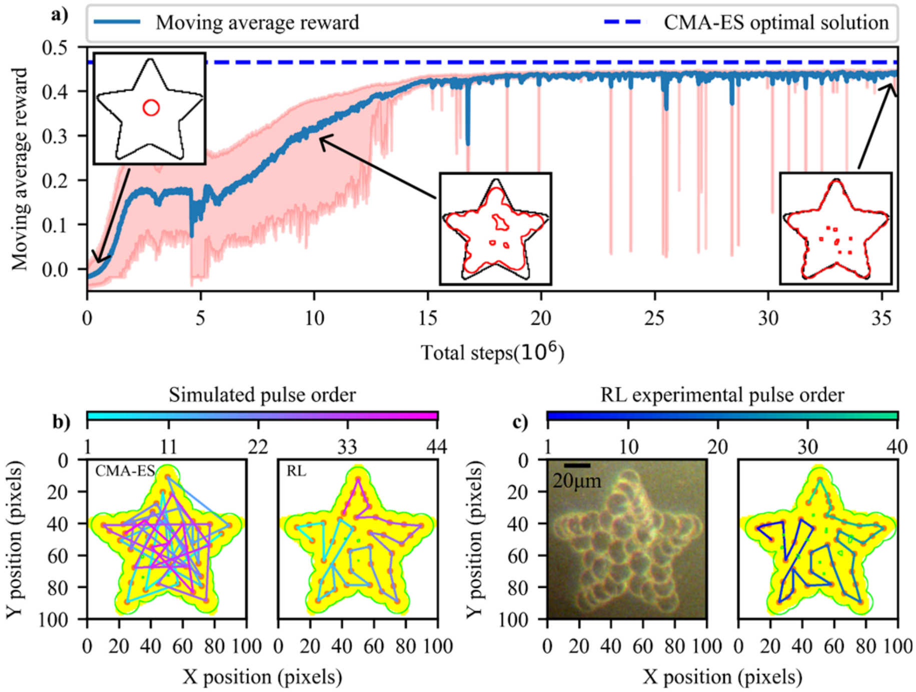

Fig. 3. a) Training curve; moving average of the reward that the RL agent obtained per training episode versus total training steps. A total number of ~1.6×10^6 episodes (i.e., full machining experiments) were conducted during this training, which was composed of ~36.0×10^6 training steps (i.e., laser pulses). The target pattern is a five-pointed star and machining performance is shown at three stages of training in the insets. b) Trajectories selected by the CMA-ES (left) and by the RL agent (right). c) The microscope image (experiment result) showing real-world laser machining controlled by the RL agent (left) and a visualization of the RL selected experimental trajectory (right).

Table 1. Comparison of CMA-ES and RL performance

Experimental target pattern type	Machining statistics	RL	CMA-ES
a) Static target pattern (section 3.1, simulation results for both RL and CMA-ES)	Target machined %	96.85	97.55
	Off-target machined %	1.72	1.93
	Number of pulses used	40	45
	Total reward	0.462	0.465
b) Static target patterns rotated to arbitrary angles (section 3.2, simulation results for both RL and CMA-ES)	Average target machined %	97.01 ± 0.54	97.44 ± 0.39
	Average off-target machined %	3.60 ± 0.80	1.92 ± 0.22
	Average total reward	0.442 ± 0.005	0.452 ± 0.002
c) Randomly generated target pattern 1 (section 3.3, experimental result for RL and simulation result for CMA-ES)	Target machined %	92.47	97.36
	Off-target machined %	3.32	1.17
	Number of pulses used	33	28
d) Randomly generated target pattern 2 (section 3.3, experimental result for RL and simulation result for CMA-ES)	Target machined %	93.05	98.19
	Off-target machined %	3.24	0.77
	Number of pulses used	39	45

conditions and/or visual changes in the vicinity of the laser machining (e.g., pressure shock waves caused by the laser pulses can blow dust and small fragments off the workpiece – resulting in visual changes that are not centered at the laser machining position). 2) Backlash, drift, or vibrations of the XY translation stages. Despite errors arising from these software and hardware limitations, the RL agent, with the help of its workpiece observations, was able to compensate and successfully machine the target pattern. This self-correction ability is investigated further in section 3.4.

3.2. Static target pattern rotated at a random angle

In this section, we increase the difficulty of the task presented to the RL agent by attempting toolpath generation for a static target pattern that can be rotated to any angle. For this experiment, the RL agent is trained in a virtual environment where the initial target pattern is again the five-pointed star (as used in section 3.1), but can be rotated about its image center by a random angle, i.e., the rotation angle is a floating-point number drawn from a uniform distribution between 0 and 360 degrees at the start of each training episode.

The training curve of the RL agent is shown in Fig. 4(a)), note that the horizontal scale is very much larger than that used in Fig. 3(a)). After training, the performance of the RL agent is assessed with target patterns at each integer angle (i.e., 360 samples in total); a degree-by-degree comparison between the RL agent and the CMA-ES method is presented in Fig. 4(b)).

From the data presented in Fig. 4(b)), a quantitative comparison between the RL and the CMA-ES results is presented in Table 1 row (b). The RL agent demonstrates target machining performance that is very close to the CMA-ES determined optimal value (within 3% in terms of reward and within 0.5% in terms of target machining percentage), but with slightly higher (although <1.9 times more) inadvertent machining of the surrounding material.

The average rewards reported here (even for the CMA-ES method) are lower than the rewards reported for a static target pattern in the previous section (i.e., Fig. 3(b))). This is a direct consequence of rotating the star-shaped target pattern which, at some angles, causes the tips of some arms of the star to move outside of the observable workpiece area. This reduces the total area of some target patterns and hence also reduces the maximum obtainable reward.

3.3. Randomly generated arbitrary target patterns

In this section, we further demonstrate that the RL agent is applicable to randomly generated, arbitrary, target patterns. Here, the RL agent is trained in a virtual environment where a random shape generator offers a near-infinite number of possible target patterns. Target patterns are defined within a 100×100 pixel image array that represents the virtual workpiece. Seven pixels are selected at random (i.e., seven pairs of XY coordinates are drawn from a uniform distribution of integers between 1 and 100), these points are then connected using a parametric curve (Bézier). In Fig. 5, nine examples of target patterns are presented, with the randomly selected points (curve nodes) highlighted in green. For seven points randomly selected on a 100×100 virtual workpiece, the total number of combinations C_k^n can be calculated using $C_k^n = n! / k!(n-k)!$ giving $C_7^{10000} \approx 1.98 \times 10^{24}$. However, it is possible for certain combinations of randomly selected points to generate the same target pattern once digitized into the image array. In addition, target patterns with small area tend to have thin features that are impossible to machine given the laser spot size; consequently, target patterns with small areas are automatically discarded. The total number of distinct target patterns that can be generated by the random shape generator was therefore also estimated using a Monte Carlo method and was found to be approximately 0.55 times C_7^{10000}. The importance of this very large number of possible target shapes is that it ensures that target shapes used during testing of the RL agent's performance have a vanishingly small likelihood of having been encountered during training.

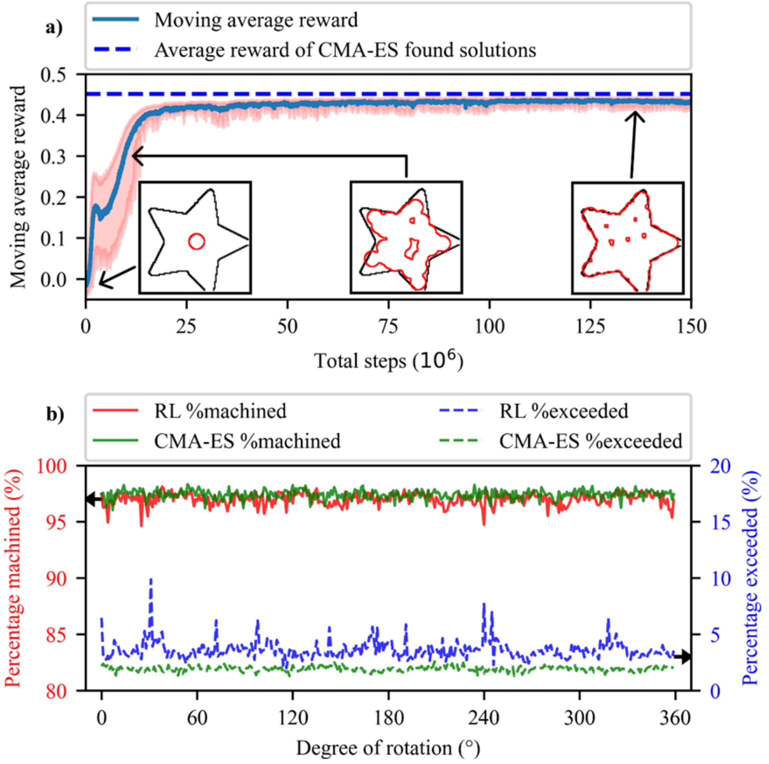

Fig. 4. a) Moving-average reward obtained by the RL agent per episode during training, versus total training steps. In this training, a total number of ~4.1×10^6 episodes (i.e., full machining experiments) were conducted, which was composed of ~150.0×10^6 training steps (i.e., laser pulses). The target pattern here is again the five-pointed star but it can occur rotated by any angle in the range 0 - 360 degrees. b) A degree-by-degree comparison showing the percentage of the target patterns successfully machined, and the percentage of inadvertent machining from the surrounding materials. Values are shown for both the RL agent (solid red and dashed blue lines), and the CMA-ES method (solid and dashed green lines).

Figure 6(a)) shows the training curve for the RL agent. The variance of the moving-average reward is much higher than it was in Fig. 4(a)). This is primarily because, here, the shape and area of the target pattern vary considerably between episodes, causing large variations in the maximum obtainable reward (for comparison, the 5th and 95th percentiles of the maximum obtainable reward are shown by the green and red dashed lines respectively). Additionally, the occurrence in the random shape generator, of narrow features that cannot be correctly machined (as noted in Fig. 5) further contributes to the observed variance. Theoretically, the CMA-ES could be employed here to search for optimal solutions for each of the target patterns generated during training, to permit the same quantitative evaluation of RL performance versus CMA-ES performance as was presented in the previous sections (the upper bound estimates shown by the horizontal lines in Fig. 3(a)) and Fig. 4(a)) required only 1 and 360 CMA-ES optimizations respectively). However, given that CMA-ES optimization for a single target pattern takes approximately 15 minutes to compute in a high-end desktop configured with a 9th generation Intel i9 processor, it would take approximately 290 years to do this for all 1×10^7 target patterns generated during training. We therefore estimate the upper bound of reward for Fig. 6(a)) in a different way, namely by calculating the average of the total reward available (based on the

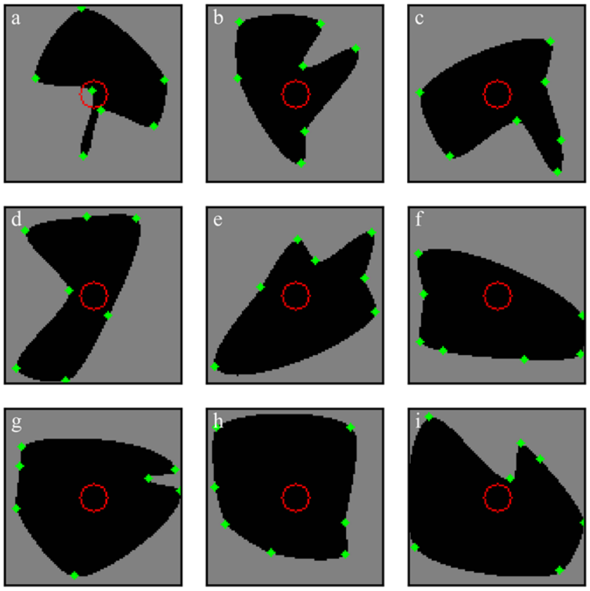

Fig. 5. a) to i) Example target patterns generated by the random shape generator. The areas from a) to i) respectively are 3078, 3375, 3759, 3800, 4370, 4520, 5419, 5743 and 6208 pixels. The 7 green markers on each target pattern are randomly selected anchor points which are connected via a parametric curve to form a closed shape. The red circles displayed on each sub-figure demonstrate the approximate size of a laser pulse trace. Evidently some features of the randomly generated shapes are too small to be correctly laser machined, most noticeably the lower part of a).

target pattern area). Note that this results in a higher upper bound than the CMA-ES based estimates shown in Fig. 3(a)) and Fig. 4(a)), as it does not take into account negative rewards that would be accrued due to unavoidable machining of background pixels in the vicinity of small features (which are considerably more likely to occur in the random target shape environment than in the fixed or rotating star environments). It is therefore understandable that the RL agent's moving-average performance shown in Fig. 6(a)) falls slightly further below the dashed line that indicates the upper bound estimate (based on target area and available reward). Nevertheless, the shape of the training curve demonstrates that the RL agent has successfully learnt to carry out this more complex, arbitrary target pattern, laser machining task.

Analysis of the RL agent performance is presented in Fig. 6(b)) in the form of a box-and-whisker diagram. After training, 140,000 random target patterns and associated laser machining trajectories were generated. This data was separated into 7 categories (by pixel area, as stated in the horizontal axis labels of Fig. 6(b))), each of which contains 20,000 target patterns and their machining trajectories. When sorted by size in this way, and by plotting the percentage machined rather than the reward (which depends upon target pattern area) much of the variability that could be observed in Fig. 6(a) is removed, and it is evident that the RL agent successfully machines >90% of the target pattern in almost all cases. Figure 6(b)) shows that the average percentage of the target that is successfully machined increases with the area of the target pattern, and the amount of inadvertent machining outside of the target decreases with the area of the target pattern. In addition, the distribution of the percentages for both successful and inadvertent machining become more symmetric as the area of a target pattern grows. It is evident that performance of the RL agent increases with the area of the target pattern; this is because for larger target patterns the RL agent has a lower probability of encountering narrow features that are impossible to machine

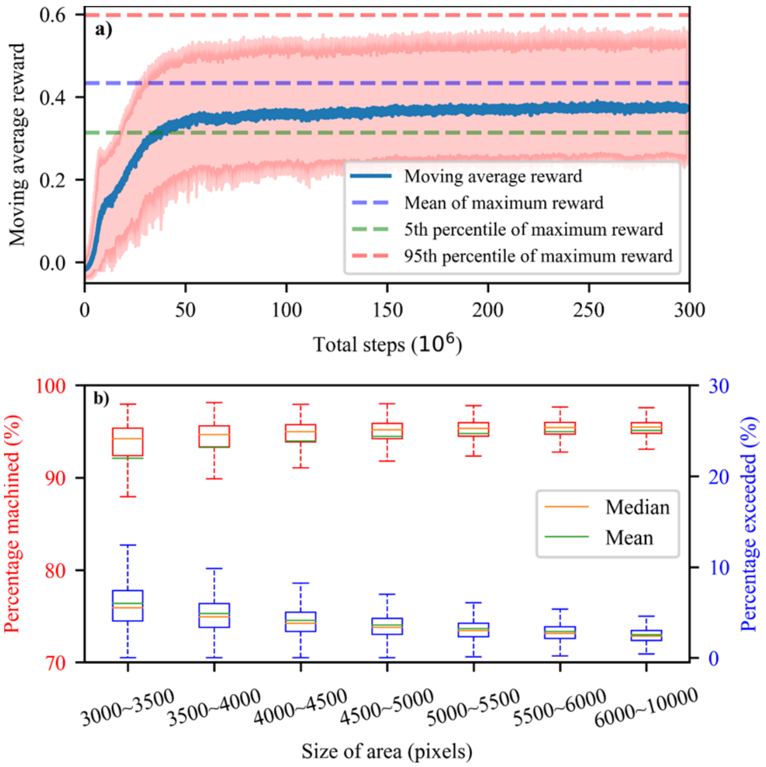

Fig. 6. a) Moving-average reward obtained by the RL agent per episode during training versus total training steps. A total number of ~10.0×10^6 episodes (i.e., full machining experiments) were conducted in this training, which was composed of ~300.0×10^6 training steps (i.e., laser pulses). The target patterns here were produced by the random shape generator during the training. b) Performance assessment of the RL agent in the form of a box-and-whisker diagram, where data are categorized by area of the target pattern. It reports the average percentage of the target pattern successfully machined (upper), and the average percentage of the target pattern area inadvertently machined from the surrounding material (lower).

correctly (also, the reward associated with any narrow features is a smaller proportion of the total available reward). Equivalently, the laser spot is proportionally smaller relative to larger target patterns and is therefore able to machine the shape more precisely. For target patterns with small area, the dimensions of its features are often smaller than the diameter of the laser-machined trace (~18 µm), machining such features would result in an insignificant amount of positive reward, sometimes even a negative reward, and therefore the RL agent is incentivized not to select XY positions in these regions.

Figure 7 shows the results of real-world experiments where the RL agent controls both the translation stages and the laser, in order to machine randomly generated target patterns. As previously stated, the number of possible distinct target patterns is enormous, and each distinct target pattern has an equal chance of being selected during training. It is therefore extremely unlikely that the exact target patterns used in these experiments were seen by the RL agent during training. It is even more unlikely that these target patterns could occur with sufficient frequency during training that the RL agent would be able to learn dedicated strategies of XY positions for these target patterns (which may have been possible for the static target pattern presented in section 3.1). These results therefore demonstrate that the RL agent has learned a generalized policy that allows it to generate toolpaths for arbitrary target patterns. A quantitative comparison between the RL agent and CMA-ES performance is presented on the right of Fig. 7 and in Table 1 row (c) and (d).

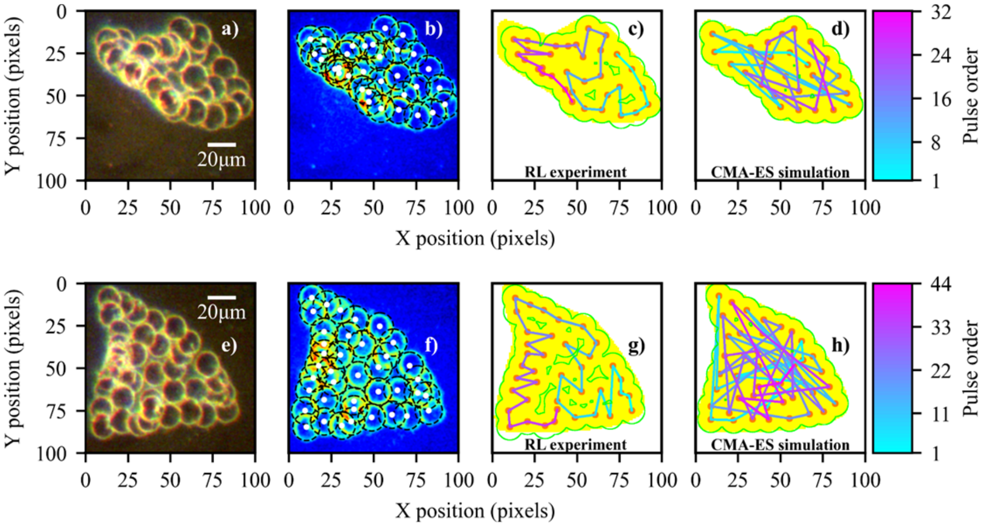

Fig. 7. a) and e) Camera images recorded at the end of the laser machining experiment for target patterns 1 and 2 respectively. b) and f) A visualization showing the outline of individual laser pulse markings (black dashed circles) and their centers (white dots) as was determined via the image processing method. c) and g) Visualizations of the initial target shape (yellow) and the laser machining trajectory chosen by the RL agent. d) and h) trajectories selected by the CMA-ES given the same generated target patterns.

The RL agent's performance values listed in Table 1 row (c) and d) are lower than those shown in the previous sections (i.e., for Fig. 3(b)) and Fig. 4(b))). This is expected because the task (machining of arbitrary target patterns) is more complex and requires the RL agent to learn a more general policy. During training in the arbitrary shape virtual environment, the RL agent only encounters a small fraction of all possible target patterns $\sim 10^7/10^{24}$, and yet, must discover a policy that functions for any of the remaining, unseen, target patterns. In contrast, in sections 3.1 and 3.2, the RL agent may repeatedly encounter the same target pattern, giving it the opportunity to further optimize its policy.

It is obvious from Table 1 row (c) and (d) and Fig. 7(d)) and h) that the RL results are slightly less good to the CMA-ES ones, however, as explained previously the CMA-ES results represent the theoretically optimal solution (or very close to it). Additionally, the results for the RL agent were obtained in the real-world experiment and are therefore subject to experimental errors (e.g., backlash) which are not present for the CMA-ES results obtained in the virtual environment. The CMA-ES results therefore define the upper limit of possible performance for the RL agent, and hence, the comparable level of performance currently reached by the RL agent may be more than sufficient for many laser machining operations. Critically, whilst the CMA-ES optimization takes 15 minutes for each new arbitrary target pattern, the computation time for the trained RL agent is significantly less than 1 second (hence a three orders of magnitude speed increase). This means that the RL approach could be applicable in high-speed processing where the CMA-ES method would be infeasible. Furthermore, additional gains in performance may be possible for the RL agent via network and hyperparameter optimization which was not undertaken for this proof-of-principle experiment.

3.4. Positional randomness and self-correction

In section 3.1, we briefly mentioned the self-correction ability of the RL agent which compensates for inadvertent errors in laser pulse positioning; in this section, we further elaborate on this ability. Fundamentally, this self-correction ability stems from the stochastic nature of the adopted RL algorithm, PPO, i.e., the policy for selecting XY positions in PPO is mathematically a conditional joint probability distribution (i.e., a 2D Gaussian). That is, the PPO algorithm optimizes a joint probability distribution towards receiving higher total reward based upon observations of the workpiece. For a given observation, the most probable XY position for the RL agent to select is the mean of the (Gaussian) distribution, however, since the distribution extends to $\pm\infty$ it is possible for all other XY positions to be selected during training. Therefore, in order for the RL agent to maximize its total reward, the strategy that it learns must be able to tolerate selection of non-preferred XY positions. In other words, due to the randomness embedded in the RL agent, it explores a large number of sub-optimal trajectories where the XY positions diverge from the optimal trajectory (i.e., the mean of the Gaussian distribution). Consequently, when employed in the physical experiment (where XY positions are always selected from the peak of the probability distribution) the RL agent is able to tolerate errors in XY pulse positioning (arising, for example, from stage vibrations) and by following a sub-optimal trajectory, is nevertheless able to successfully complete the laser machining task.

Figure 8 demonstrates the self-correction ability of the RL agent that was presented in section 3.3 for an elliptical target pattern and a "thumbs-up" target pattern. In these demonstrations, the 5th laser pulses (highlighted in red) were deliberately displaced (from their undisturbed locations in Fig. 8(a)) and (d)) to all possible positions on their respective virtual workpieces. In almost all cases the RL agent was able to successfully complete machining of the target pattern regardless of this disturbance (see supplementary Visualization 1 and Visualization 2 for the elliptical target pattern and "thumbs-up" target pattern respectively). Figure 8(b)), c), e) and f) show

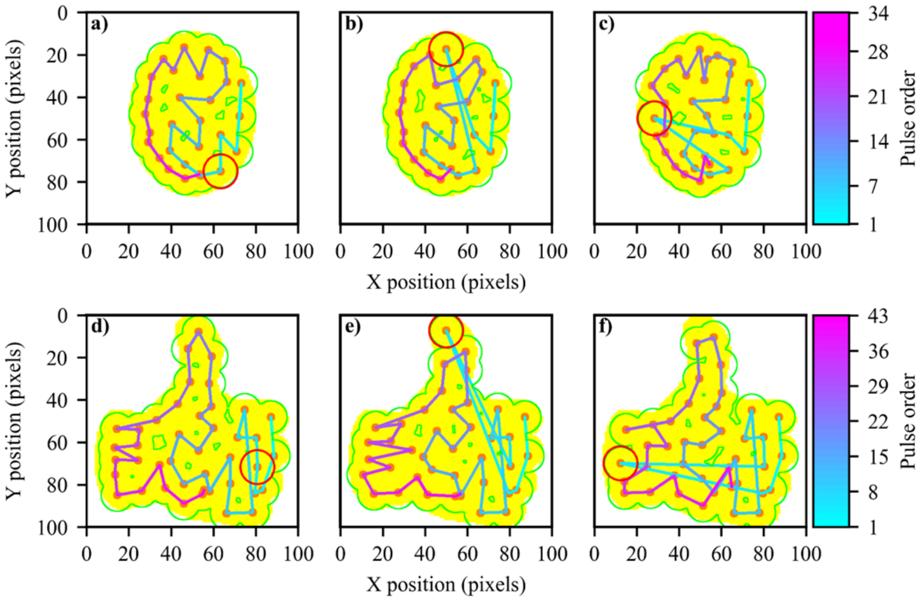

Fig. 8. a) and d) Simulations showing the RL agent's preferred trajectories for an ellipse shaped target pattern and a "thumbs-up" target pattern respectively, where the 5th laser pulse positions are highlighted with a red circle. b), c), e) and f) Simulations of the RL agent's trajectory when the 5th laser pulse is deliberately displaced.

example trajectories where the 5th laser pulse has been deliberately displaced. In addition, the self-correction ability can be observed in other target patterns shown in this work. Specifically, the self-correction ability for the target patterns presented in Fig. 1, Fig. 7(a)) and e) are shown in supplementary Visualization 3, Visualization 4, and Visualization 5, respectively.

The RL agent makes observations of the workpiece at each step and therefore has the opportunity to correct for any incorrectly positioned pulses. In contrast, the CMA-ES algorithm calculates the complete machining trajectory based only on the initial target pattern and without making any subsequent observation of the workpiece. In this implementation, it is therefore impossible for the CMA-ES algorithm to offer the capability for automatic error correction that is inherent within the RL PPO method.

4. Conclusions

In this work, a novel method for laser machining has been demonstrated; facilitated by reinforcement-learning-control of a laser and translation stages. RL agents were assigned with performing autonomous toolpath generation tasks of increasing complexity from static target patterns, static target patterns rotated to arbitrary angles and randomly generated arbitrary target patterns. The RL agents were able to learn each of these tasks and achieved comparable performance to the theoretically optimal solution (which was estimated using the CMA-ES optimization algorithm).

Unlike search-based methods, such as CMA-ES, where the entire toolpath is determined prior to machining, the proposed RL approach deduces only the position at which the next laser pulse should be applied (with reference to observations of a workpiece made in real time). Feedback via these workpiece observations, and the PPO algorithm's inherent tolerance to variance in the executed action, bestow the RL agent with the ability to automatically correct for machining errors (for example those arising from stage vibrations). This capability was demonstrated in the virtual environment by deliberately displacing selected laser pulses relative to their intended locations and the RL agent proved to be robust against this form of interference. Further evidence of this self-correction capability is provided by small differences between trajectories, for the same target shape, that were observed in virtual and real-world experiments (such corrections only being necessary and present in the real-world case).

The most important result here is that it was possible for the RL agent to learn a policy, within the virtual environment, that allows it to laser machine arbitrary target shapes in the real-world experiment. Once trained in this manner, the RL agent can design a toolpath for a target shape extremely quickly and with only minimal computational requirements. The RL approach therefore offers potential benefits in high-speed laser machining applications (where traditional iterative optimization methods would be infeasible) and could be particularly suitable in cases such as automated rapid prototyping, where a large number of different target patterns could be encountered or when target patterns may not even be known in advance.

Funding. Engineering and Physical Sciences Research Council (EP/N03368X/1, EP/T026197/1).

Acknowledgments. B.M. was supported by the Engineering and Physical Research Council Early Career Fellowships Scheme (EP/N03368X/1). This work was supported by the Engineering and Physical Research Council under grant number EP/T026197/1.

Disclosures. The authors declare no conflicts of interest.

Data availability. Data underlying the results presented in this paper are available in Ref. [51].

Supplemental document. See Supplement 1 for supporting content.

References

1. A. K. Dubey and V. Yadava, "Laser beam machining—A review," Int. J. Mach. Tools Manuf. **48**(6), 609–628 (2008).
2. B. Dusser, Z. Sagan, H. Soder, N. Faure, J. P. Colombier, M. Jourlin, and E. Audouard, "Controlled nanostructrures formation by ultra fast laser pulses for color marking," Opt. Express **18**(3), 2913–2924 (2010).

3. A. Mahrle and E. Beyer, “Theoretical aspects of fibre laser cutting,” J. Phys. D: Appl. Phys. **42**(17), 175507 (2009).
4. K. M. Nowak, H. J. Baker, and D. R. Hall, “Efficient laser polishing of silica micro-optic components,” Appl. Opt. **45**(1), 162–171 (2006).
5. D. Bhaduri, P. Penchev, A. Batal, S. Dimov, S. L. Soo, S. Sten, U. Harrysson, Z. Zhang, and H. Dong, “Laser polishing of 3D printed mesoscale components,” Appl. Surf. Sci. **405**, 29–46 (2017).
6. E. O. Olakanmi, R. F. Cochrane, and K. W. Dalgarno, “A review on selective laser sintering/melting (SLS/SLM) of aluminium alloy powders: Processing, microstructure, and properties,” Prog. Mater. Sci. **74**, 401–477 (2015).
7. M. Malinauskas, A. Žukauskas, S. Hasegawa, Y. Hayasaki, V. Mizeikis, R. Buividas, and S. Juodkazis, “Ultrafast laser processing of materials: from science to industry,” Light: Sci. Appl. **5**(8), e16133 (2016).
8. E. Kannatey-Asibu Jr., *Principles of Laser Materials Processing* (Wiley, 2009).
9. J. Dutta Majumdar and I. Manna, “Laser material processing,” Int. Mater. Rev. **56**(5-6), 341–388 (2011).
10. D. Bäuerle, *Laser Processing and Chemistry* (Springer Berlin Heidelberg, 2011).
11. X. Liu, D. Du, and G. Mourou, “Laser ablation and micromachining with ultrashort laser pulses,” IEEE J. Quantum Electron. **33**(10), 1706–1716 (1997).
12. Y. Xie, D. J. Heath, J. A. Grant-Jacob, B. S. Mackay, M. D. T. McDonnell, M. Praeger, R. W. Eason, and B. Mills, “Deep learning for the monitoring and process control of femtosecond laser machining,” J. Phys. Photonics **1**(3), 035002 (2019).
13. C. Momma, S. Nolte, B. N. Chichkov, F. V. Alvensleben, and A. Tünnermann, “Precise laser ablation with ultrashort pulses,” Appl. Surf. Sci. **109-110**, 15–19 (1997).
14. Z. Sun and J. C. Ion, “Laser welding of dissimilar metal combinations,” J. Mater. Sci. **30**(17), 4205–4214 (1995).
15. F. Bachmann and U. Russek, *Laser welding of polymers using high-power diode lasers*, Laser Processing of Advanced Materials and Laser Microtechnologies (SPIE, 2003), Vol. 5121.
16. A. Matsunawa, J.-D. Kim, N. Seto, M. Mizutani, and S. Katayama, “Dynamics of keyhole and molten pool in laser welding,” J. Laser Appl. **10**(6), 247–254 (1998).
17. T. C. Chong, M. H. Hong, and L. P. Shi, “Laser precision engineering: from microfabrication to nanoprocessing,” Laser Photonics Rev. **4**(1), 123–143 (2010).
18. T. Podżorny, G. Budzyń, and J. Rzepka, “Linearization methods of laser interferometers for pico/nano positioning stages,” Optik **124**(23), 6345–6348 (2013).
19. K. Sugioka and Y. Cheng, “Femtosecond laser processing for optofluidic fabrication,” Lab Chip **12**(19), 3576–3589 (2012).
20. D.-P. Wan, X.-C. Liang, F.-M. Meng, D.-J. Hu, Y.-M. Wang, B.-K. Chen, and Y.-M. Shao, “Automatic compensation of laser beam focusing parameters for flying optics,” Opt. Laser Technol. **41**(4), 499–503 (2009).
21. A. Žemaitis, M. Gaidys, P. Gečys, G. Račiukaitis, and M. Gedvilas, “Rapid high-quality 3D micro-machining by optimised efficient ultrashort laser ablation,” Opt. Lasers Eng. **114**, 83–89 (2019).
22. D. J. Heath, T. H. Rana, R. A. Bapty, J. A. Grant-Jacob, Y. H. Xie, R. W. Eason, and B. Mills, “Ultrafast multi-layer subtractive patterning,” Opt. Express **26**(9), 11928–11933 (2018).
23. M. Jiang, X. Wang, S. Ke, F. Zhang, and X. Zeng, “Large scale layering laser surface texturing system based on high speed optical scanners and gantry machine tool,” Robot. Comput. Integr. Manuf. **48**, 113–120 (2017).
24. M. L. Tseng, P. C. Wu, S. Sun, C. M. Chang, W. T. Chen, C. H. Chu, P. L. Chen, L. Zhou, D. W. Huang, T. J. Yen, and D. P. Tsai, “Fabrication of multilayer metamaterials by femtosecond laser-induced forward-transfer technique,” Laser Photonics Rev. **6**(5), 702–707 (2012).
25. S. Prakash and S. Kumar, “Pulse smearing and profile generation in CO2 laser micromachining on PMMA via raster scanning,” Journal of Manufacturing Processes **31**, 116–123 (2018).
26. T. H. C. Childs and C. Hauser, “Raster scan selective laser melting of the surface layer of a tool steel powder bed,” Proc. Inst. Mech. Eng., Part B **219**(4), 379–384 (2005).
27. H. Tünnermann and A. Shirakawa, “Deep reinforcement learning for tiled aperture beam combining in a simulated environment,” J. Phys. Photonics **3**(1), 015004 (2021).
28. N. Bruchon, G. Fenu, G. Gaio, M. Lonza, F. H. O’Shea, F. A. Pellegrino, and E. Salvato, “Basic Reinforcement Learning Techniques to Control the Intensity of a Seeded Free-Electron Laser,” Electronics **9**(5), 781 (2020).
29. M. Praeger, Y. Xie, J. A. Grant-Jacob, R. W. Eason, and B. Mills, “Playing optical tweezers with deep reinforcement learning: in virtual, physical and augmented environments,” Mach. Learn.: Sci. Technol. **2**(3), 035024 (2021).
30. J. Günther, P. M. Pilarski, G. Helfrich, H. Shen, and K. Diepold, “Intelligent laser welding through representation, prediction, and control learning: An architecture with deep neural networks and reinforcement learning,” Mechatronics **34**, 1–11 (2016).
31. D. Ha and J. Schmidhuber, “Recurrent World Models Facilitate Policy Evolution,” S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds. (2018).
32. G. Masinelli, T. Le-Quang, S. Zanoli, K. Wasmer, and S. A. Shevchik, “Adaptive Laser Welding Control: A Reinforcement Learning Approach,” IEEE Access **8**, 103803–103814 (2020).
33. Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel, “Benchmarking Deep Reinforcement Learning for Continuous Control,” in *Proceedings of The 33rd International Conference on Machine Learning*, B. Maria Florina and Q. W. Kilian, eds. (PMLR, Proceedings of Machine Learning Research, 2016), pp. 1329–1338.
34. T. N. Larsen, H. Ø. Teigen, T. Laache, D. Varagnolo, and A. Rasheed, “Comparing Deep Reinforcement Learning Algorithms’ Ability to Safely Navigate Challenging Waters,” Front. Robot. AI **8**, 738113 (2021).

35. P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, “Deep Reinforcement Learning That Matters,” *Proceedings of the AAAI Conference on Artificial Intelligence* **32** (2018).
36. A. Hill, A. Raffin, M. Ernestus, A. Gleave, A. Kanervisto, R. Traore, P. Dhariwal, C. Hesse, O. Klimov, A. Nichol, M. Plappert, A. Radford, J. Schulman, S. Sidor, and Y. Wu, “Stable Baselines,” (GitHub, https://github.com/hill-a/stable-baselines, 2018).
37. V. Mnih, A. Puigdomènech Badia, M. Mirza, A. Graves, T. P. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, “Asynchronous Methods for Deep Reinforcement Learning,” (2016), p. arXiv:1602.01783.
38. T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, “Continuous control with deep reinforcement learning,” arXiv preprint arXiv:1509.02971 (2015).
39. J. Ho and S. Ermon, “Generative Adversarial Imitation Learning,” (2016), p. arXiv:1606.03476.
40. J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal Policy Optimization Algorithms,” (2017), p. arXiv:1707.06347.
41. T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, “Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor,” (2018), p. arXiv:1801.01290.
42. S. Fujimoto, H. Van Hoof, and D. Meger, “Addressing function approximation error in actor-critic methods,” arXiv preprint arXiv:1802.09477 (2018).
43. J. Schulman, S. Levine, P. Moritz, M. I. Jordan, and P. Abbeel, “Trust Region Policy Optimization,” (2015), p. arXiv:1502.05477.
44. I. G. B. Petrazzini and E. A. Antonelo, “Proximal Policy Optimization with Continuous Bounded Action Space via the Beta Distribution,” (2021), p. arXiv:2111.02202.
45. J. Oh, Y. Guo, S. Singh, and H. Lee, “Self-Imitation Learning,” in *Proceedings of the 35th International Conference on Machine Learning*, D. Jennifer and K. Andreas, eds. (PMLR, Proceedings of Machine Learning Research, 2018), pp. 3878–3887.
46. O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, J. Oh, D. Horgan, M. Kroiss, I. Danihelka, A. Huang, L. Sifre, T. Cai, J. P. Agapiou, M. Jaderberg, A. S. Vezhnevets, R. Leblond, T. Pohlen, V. Dalibard, D. Budden, Y. Sulsky, J. Molloy, T. L. Paine, C. Gulcehre, Z. Wang, T. Pfaff, Y. Wu, R. Ring, D. Yogatama, D. Wünsch, K. McKinney, O. Smith, T. Schaul, T. Lillicrap, K. Kavukcuoglu, D. Hassabis, C. Apps, and D. Silver, “Grandmaster level in StarCraft II using multi-agent reinforcement learning,” Nature **575**(7782), 350–354 (2019).
47. A. Raffin, “RL Baselines Zoo,” in *GitHub repository*, (GitHub, 2018).
48. G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, “OpenAI gym,” arXiv preprint arXiv:1606.01540 (2016).
49. N. Hansen, “The CMA Evolution Strategy: A Comparing Review,” in *Towards a New Evolutionary Computation: Advances in the Estimation of Distribution Algorithms*, J. A. Lozano, P. Larrañaga, I. Inza, and E. Bengoetxea, eds. (Springer, Berlin, Heidelberg, 2006), pp. 75–102.
50. N. Hansen, “The CMA evolution strategy: A tutorial,” arXiv preprint arXiv:1604.00772 (2016).
51. Y. Xie, M. Praeger, J. A. Grant-Jacob, R. W. Eason, and B. Mills, “Data for ‘Motion control for laser machining via reinforcement learning,” (2022), retrieved.