



This is a repository copy of *Beat the machine (learning) : metal additive manufacturing and closed loop control*.

White Rose Research Online URL for this paper:
<http://eprints.whiterose.ac.uk/163209/>

Version: Published Version

Article:

Freeman, F.S.H.B. orcid.org/0000-0002-7402-1790, Chechik, L. and Todd, I. (2020) Beat the machine (learning) : metal additive manufacturing and closed loop control. *Physics Education*, 55 (5). 055012. ISSN 0031-9120

<https://doi.org/10.1088/1361-6552/ab9957>

Reuse

This article is distributed under the terms of the Creative Commons Attribution (CC BY) licence. This licence allows you to distribute, remix, tweak, and build upon the work, even commercially, as long as you credit the authors for the original work. More information and the full terms of the licence here:
<https://creativecommons.org/licenses/>

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.



eprints@whiterose.ac.uk
<https://eprints.whiterose.ac.uk/>

PAPER • OPEN ACCESS

Beat the machine (learning): metal additive manufacturing and closed loop control

To cite this article: Felicity S H B Freeman *et al* 2020 *Phys. Educ.* **55** 055012

View the [article online](#) for updates and enhancements.



Physics of Life Online An interdisciplinary course

Physics of Life Online is an innovative exploration of the interface between physics and the phenomenon of life. The course is aimed at undergraduate students in the sciences, taught from the perspective of a biologist.

Physics of Life Online consists of four modules:

- **Thermodynamics** The thermodynamics of life, its function and its origin.
- **Biomechanics** Engineering life through innovative structures and materials.
- **Fluids** Life lives in fluids. How does it manage the fluid environment?
- **Wave phenomena** Light and sound are life's means of sensing and navigating the world.

Each module is equivalent to a 1-credit college course. Students can earn academic credit through their home institutions using independent study credits. All four modules are available through [udemy.com](#). Search: "Physics of Life". Sample episodes are available.

Free sample episodes and more are available on [YouTube](#). Search: Turner: "Physics of Life".

Scott Turner is Emeritus Professor of Biology (SUNY ESF). His interests are in the physics of adaptation and evolution.

Click to find out more or visit: <https://jscottturner.com>

Beat the machine (learning): metal additive manufacturing and closed loop control

Felicity S H B Freeman^{ORCID}, Lova Chechik^{ORCID} and Iain Todd

Department of Materials Science & Engineering, University of Sheffield, Sheffield S1 3JD, United Kingdom

E-mail: f.freeman@sheffield.ac.uk



Abstract

3D printing (additive manufacturing) is an emerging technology with the ability to make complex, free-form shapes from materials including plastics, metals and ceramics. While additive manufacturing has many advantages over more traditional processes, it can be difficult to control, which can then lead to defects in the finished part. Closed-loop control is a key part of most modern manufacturing and household processes, improving efficiency and reducing variation. Machine learning is an extension of this, where the controller learns how changes in the input variables affect the output. Here we provide an overview of the different types of metal additive manufacturing processes, and their relative strengths and weaknesses. We also describe how closed-loop control and thermal cameras are being used to improve these processes. Finally, we provide a link to a free-to-download app which allows students to control their own simulation of an additive manufacturing build, and see first-hand the need for control algorithms. Pseudo-code is provided in an appendix to help students who wish to take this further by building their own control algorithms.

Keywords: additive manufacturing, simulation, closed-loop control, educational game

1. Introduction to additive manufacturing

Alloy wheels are made by either forging or casting to an approximate shape, and then machining down to the finished size [1, 2]. This traditional approach, making a component oversize and then removing material by machining (cutting, milling, turning), is known as subtractive manufacturing (figure 1). Subtractive manufacturing is wasteful



Original content from this work may be used under the terms of the [Creative Commons Attribution 4.0 licence](https://creativecommons.org/licenses/by/4.0/). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

of both time and resources, particularly as the removed material may be contaminated in the machining process and cannot then be easily recycled.

Traditional manufacturing processes also place constraints on the geometries that can be designed. For example, forged components cannot have re-entrant features, because it would not be possible to remove the forging dies. These constraints have been challenged, particularly in the aerospace industry with advances in casting technology, but they remain a significant barrier. One common way to circumvent the geometry constraint has been to make components up from an assembly of smaller components, but this introduces additional processing steps requiring further equipment and operator time, and often adds weight to the finished part.

The concept of additive, rather than subtractive, manufacturing emerged in the early 1980's with the development of systems that used ultraviolet light to cure and harden a photosensitive polymer, building up a part layer-by-layer [3]. These evolved to the desktop polymer 3D printers that are widely available today; instead of using photosensitive polymer, these use a thermoplastic polymer filament that is fed into a heated nozzle and melted and then the molten polymer is squirted out. These also work on a layer-by-layer concept, allowing complex geometries to be achieved including re-entrant and articulated features.

By the 1990's, additive manufacturing had started to be applied to metals, and this is now a large area of research with applications in a number of high-value industries including aerospace and automotive. Additive manufacturing has been widely adopted for prototyping new designs and for building bespoke components (e.g. personalised prosthetics and implants), but is also being used for volume production manufacturing [4, 5].

The most obvious benefit of additive manufacturing is the ability to build complex shapes which would be impossible with conventional manufacturing [6–8]. It also allows weight saving as an assembly of smaller parts can be replaced by a single part with no need for joining features [9], and solid blocks of material can be replaced by open lattice structures [10]. There is also the opportunity to change material during a build,

to target the physical properties to the specific requirements of different regions [11].

There are now a large number of different additive manufacturing processes available for metals, polymers and even ceramics [12, 13]. While the terminology used can be rather confusing, all of the systems start with designing the part on a computer which splits it into layers. The thickness of the layers depends on the exact process, but can be as small as 0.01 mm. The layer geometry is loaded into the machine, and the operator then selects build settings—as a minimum these include the power and travel speed of the heat source—and starts building. Depending on the process and the geometry, a build can take as little as a few minutes or up to a full day.

2. Metal additive manufacturing

The metal additive manufacturing systems can be split into two approaches: blown-powder and powder-bed. They are both widely used, and have different strengths and weaknesses. Both use very fine metal powders, only 0.01–0.10 mm diameter (comparable with flour [14]), although similar processes exist using metal wire.

2.1. Blown powder

The blown-powder approach uses a laser beam as the heat source, focussed to a 1 mm spot diameter. The laser is directed down a cone-shaped nozzle, with metal powder flowing down a concentric channel into the focal point of the laser as shown in figure 2(a). The laser melts the powder, depositing a bead (0.2–0.4 mm high) on the substrate surface. This is comparable with polymer additive manufacturing using a glue gun or desktop polymer 3D printer. Blown-powder systems are suitable for building freeform geometry from a continuous line of material, particularly thin walled structures, and can be used for repair of damaged components (figures 2(b)–(d)).

2.2. Powder bed

The powder-bed process also generally uses a laser beam heat source, with a 0.1 mm spot diameter, although there are also systems that use electron beams. The powder-bed process uses a

Beat the machine (learning): metal additive manufacturing and closed-loop control

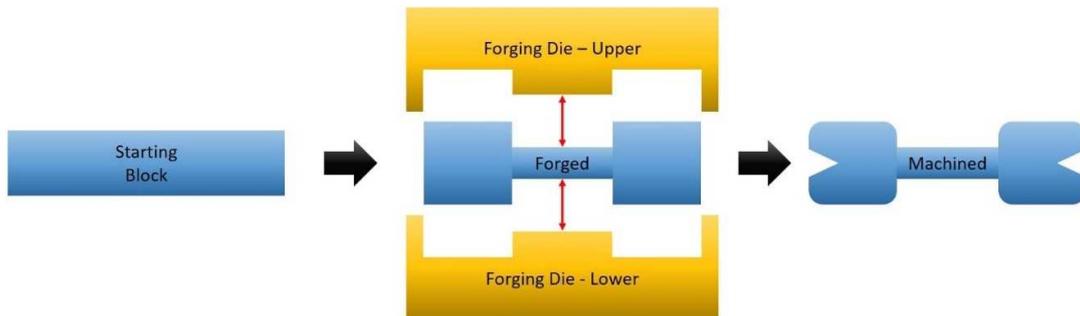


Figure 1. Schematic of a traditional subtractive manufacturing process where the block is initially forged to create the thinner middle section and wider end sections and then machined to create the indents and round the corners.

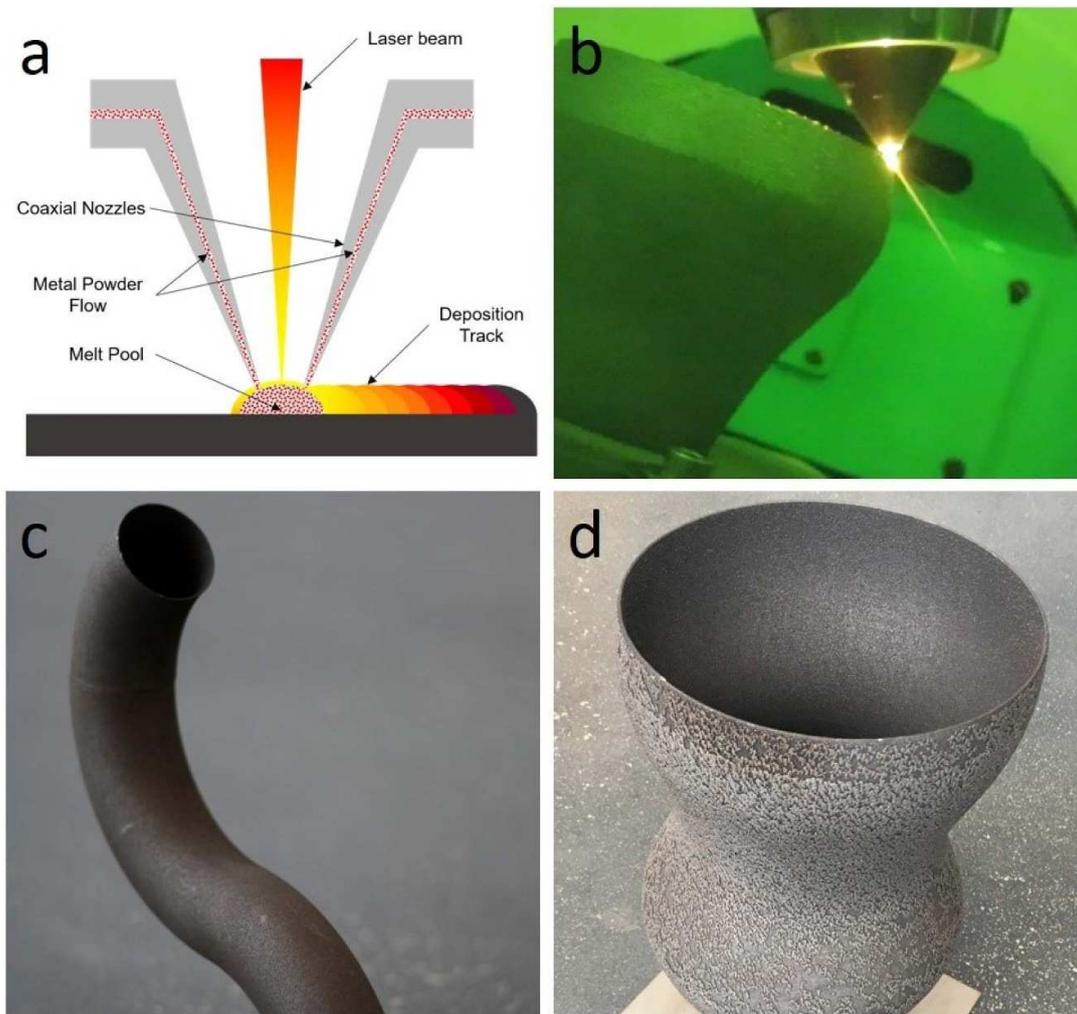


Figure 2. (a) Schematic of blown-powder direct energy deposition process; (b) blown-powder process in operation, showing nozzle and melt pool; (c) curved pipe built in titanium alloy (Ti-6Al-4 V) using blown-powder process; (d) vase built in stainless steel using blown-powder process.

layer thickness of only 0.02–0.04 mm. The first layer of powder is spread across a metal baseplate and the laser beam is used to selectively melt specific regions. Once a layer is complete, the baseplate is moved down by the layer thickness, a new layer of powder is spread on top, the laser again melts specific regions and the process repeats. The powder-bed approach is particularly suited to highly detailed components, and can be used to make articulated interlinking features (figure 3).

3. Need for control

One difficulty with metal additive manufacturing is the requirement to balance the amount of energy being supplied to the material. Too much heat (high power, low speed) and the material reaches boiling point [15]; this may cause bubbles to form inside the molten region, which can then get trapped when the metal around them solidifies (figure 4(a)). Too little heat (low power, high speed) and the powder particles do not fuse together [16]; this can leave irregular voids running through the material (figure 4(b)). Both of these are undesirable because they will cause the component to have reduced mechanical properties, and may result in it failing prematurely in operation.

Blown-powder and powder-bed both work on layer-by-layer additions of material, with the laser moving backwards and forwards across the surface, melting material in a series of hatches (lines). Where the hatches are very long, the material may cool down so much between hatches that the overall temperature is too low, and there is a risk of poor adhesion. Where these hatches are short, the residual heat from the previous hatch will not have time to fully disperse before the laser arrives on the next hatch, causing the bulk temperature to increase with a risk of boiling. There can also be a small temperature increase when the laser turns around at the end of a hatch, as it scans material that has just been melted.

For some components it may be impossible to find a single combination of power and speed which will result in a constant temperature throughout. This is illustrated in figure 5, where a fixed speed and power have been optimised for the ‘medium-length’ hatches in the middle of the layer. But using these settings for the short hatches

at the top of the triangle has resulted in boiling (red) while for the long hatches at the base it has resulted in poor adhesion (blue).

As the complexity of the geometry increases, so does the difficulty of finding a good combination of build settings. It may be necessary to run many iterations, and produce a number of ‘failed’ components before finding optimised settings. This is wasteful of both time and materials, reducing the benefits of using additive manufacturing.

Consider a self-driving car, where the two main inputs are the speed of travel (determined by the accelerator) and the direction of travel (determined by the steering wheel), but both must be set at the start of the journey and cannot be changed. For a straight road, it would be relatively easy to run a few iterations and optimise the accelerator position and steering wheel angle to achieve a reasonable journey time without collisions.

However, it would not then be possible to translate those same settings to a winding country road. Each new road would require individual trials to work out the correct settings, and for some roads it might be impossible to find just one combination of speed and direction that would work for the whole journey. A better approach would be to use on-board monitoring systems (e.g. cameras) to view the road and adjust the speed and direction as required.

This is exactly what is being developed for additive manufacturing, using built-in monitoring systems to measure the temperature of the component and adjust the power and speed as necessary [17]. The general approach is to fit thermal cameras around the component being built to monitor the temperature and shape of the melt pool. The data is processed through an algorithm to identify if it is within a target processing range. If it is outside the range, or approaching the limits, then the algorithm will adjust the laser power and/or travel speed to correct the behaviour and feed these back into the machine.

This approach is commonly called ‘closed-loop control’: a controller uses measurement equipment to read the actual value (e.g. temperature) and compares it with a reference value, if the actual value is not equal to the reference, the controller then changes the process settings by a

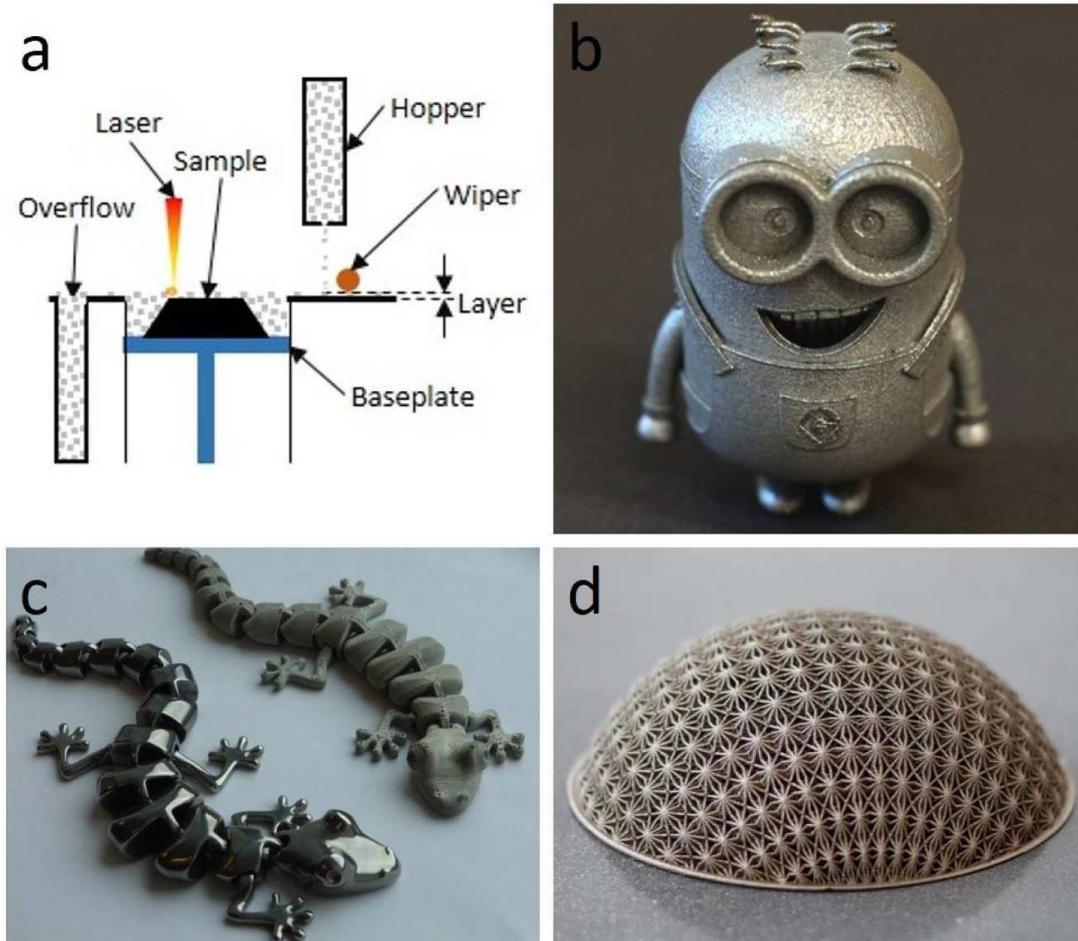


Figure 3. (a) Schematic of laser powder-bed process; (b) minion built in stainless steel using laser powder-bed; (c) articulated lizards built in stainless steel using laser powder-bed, one in as-built condition and one polished; (d) lattice mesh built in titanium alloy (Ti-6Al-4 V) using electron-beam powder bed process.

pre-programmed amount, and this should steer the actual value back towards the reference. A further development of this is ‘machine learning’, where the controller learns how changes to the input settings affect the output, so it can make finer adjustments and achieve faster convergence.

Closed-loop control is widely used across many different processes, both industrial and household. For example, domestic heating systems use it to maintain the desired room temperature (figure 6). Gaining an understanding of closed-loop control will be useful for STEM students, particularly if they are working on their own computer-controlled processes.

4. Operation

To help students visualise the additive manufacturing process, and how closed-loop control works in reality, we have created a free-to-download interactive app, available from mapp.acuk/get-involved/beat-the-machine-learning.

The app is written in Matlab App Designer version 2019a. It is packaged with Matlab Runtime, which is a free-to-download version of Matlab with the ability to run existing scripts, but cannot be used for editing. Matlab Runtime will be automatically installed as part of the app installation, but is also available from the Matlab website [18].

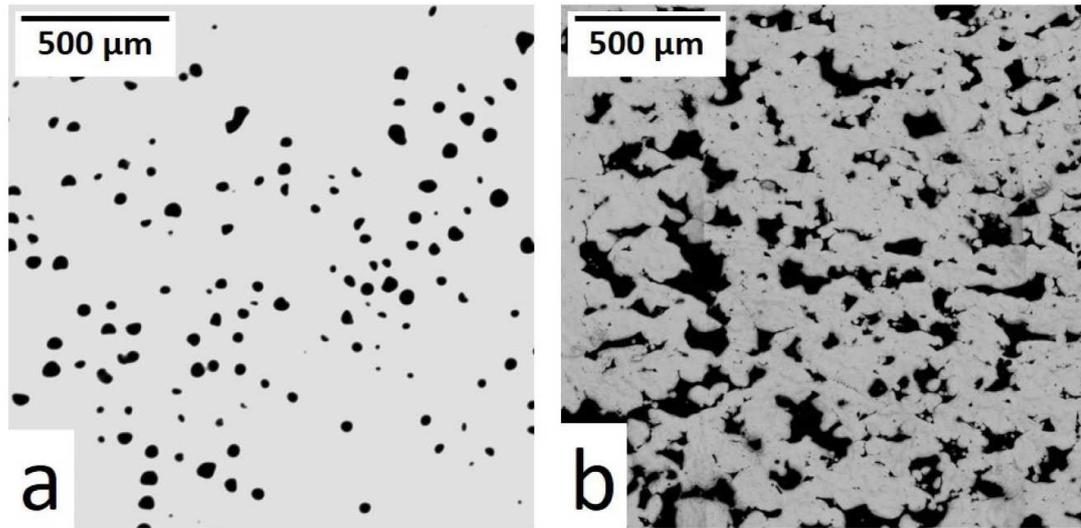


Figure 4. (a) ‘Keyhole’ porosity caused by the melt pool boiling during a powder-bed build in stainless steel; (b) ‘lack-of-fusion’ porosity caused by insufficient heat input during a powder-bed build in stainless steel.

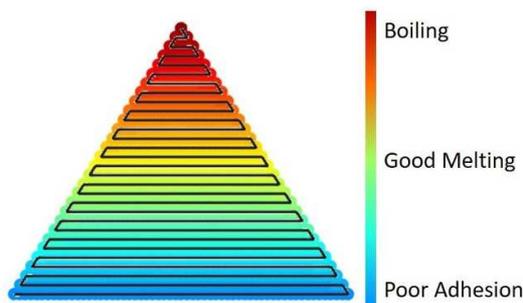


Figure 5. Schematic of temperature rise caused by changing hatch length for fixed build settings (red = hot, blue = cold) with the hatches shown in black.

There are two versions available: one controlled through a PC keyboard, and one controlled by two joysticks on an Arduino. The keyboard version with instructions can be downloaded from the MAPP website (mapp.acuk), the Arduino version is available on request from the authors. Currently they are only available for computers running Windows 10 (64-bit).

At the top of the app screen (figure 7) there is a short background explaining the need to control melt pool temperature during a build, and examples of both boiling (keyhole) porosity and poor adhesion (lack-of-fusion) porosity. The app

has radio buttons to select one of three available component geometries (square, circle, triangle), and buttons to select one of three run styles (baseline, game-mode, closed-loop control).

The left hand graph shows a plan view of the build proceeding, with the hatches shown as black lines and the colour indicating the temperature of the melt pool. The right hand graph shows a line graph of temperature against build time, with horizontal dotted lines indicating temperature limits for boiling, good melting and poor adhesion (lack-of-fusion). Using the closed-loop control terminology from above, the good melting zone between the two horizontal dotted lines is the reference value, while the actual value is the temperature indicated by the point colour in the left hand graph and by position on the y-axis in the right hand graph. The power and speed are the two settings which can be used to influence the temperature.

Selecting ‘Baseline’ will run a condition with a fixed power and speed for the whole layer of the selected geometry. At the end of the layer, the app shows a score for the percentage of time spent in the ‘good melting’ zone.

Selecting ‘Start’ puts the app into game mode, where the power and speed can be controlled by the user. If running the keyboard version, the power is controlled by Q and A, while

Beat the machine (learning): metal additive manufacturing and closed-loop control

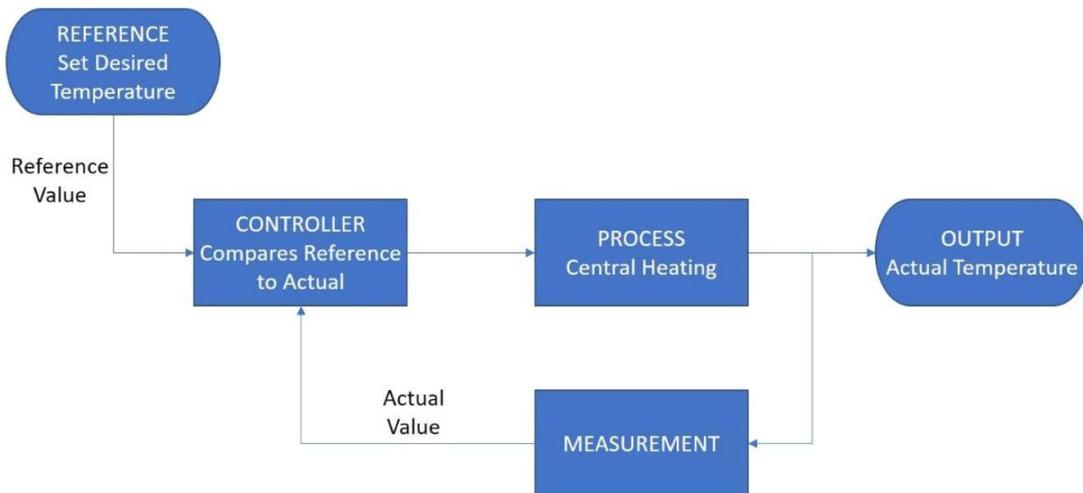


Figure 6. Flowchart showing how closed-loop control is applied to a domestic central heating system.

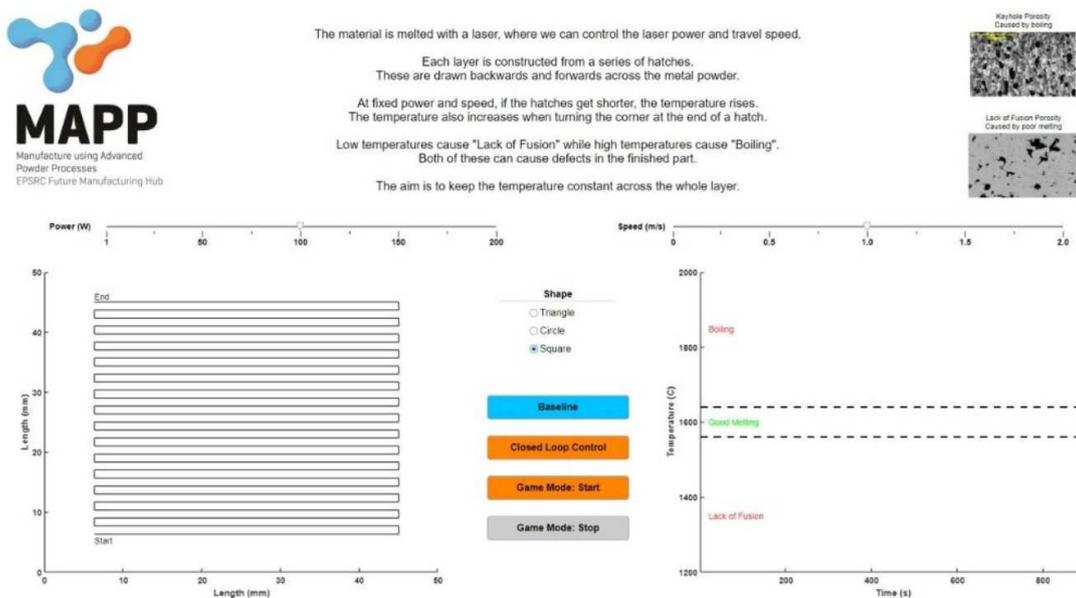


Figure 7. Opening screen of app showing introductory text, radio buttons for geometry selection, power and speed sliders and buttons for baseline, closed-loop control or game mode operation.

the speed is controlled by P and L. If running the Arduino version, both are controlled through joysticks. The app counts down and then the build starts. The user can view the live temperature data on the two graphs, and adjust the power and speed accordingly (figure 8(a)). At the end they again receive a score for the percentage of time spent in the 'good melting' zone.

Selecting 'Closed Loop Control' puts the app into an algorithm-controlled mode (figure 8(b)). This uses a fixed speed and power initially, but then adjusts according to the temperature data output. This is a similar approach to the user-controlled version, but using a designed algorithm with a much faster response rate.

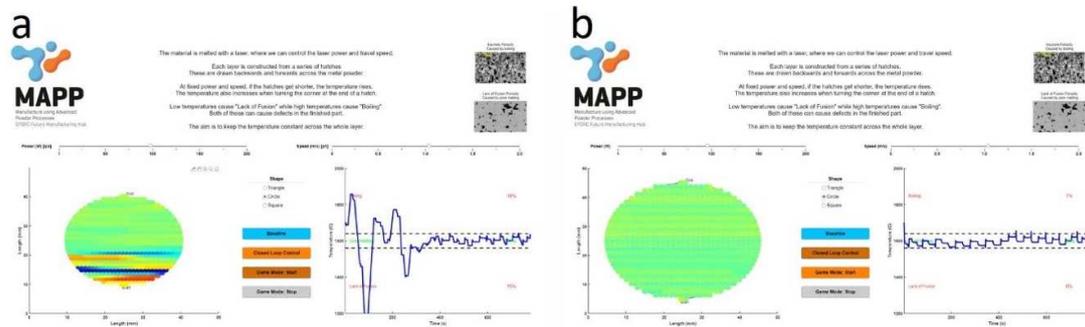


Figure 8. (a) Game mode version of circle showing poor control at the start but then improving; (b) closed loop control for the same geometry showing temperature maintained within the good melting zone across the whole geometry.

The app is designed to provide students with a general introduction to metal additive manufacturing and closed-loop control. It is also intended to demonstrate some of the challenges faced when trying to optimise a manufacturing process, and how researchers are addressing these challenges.

5. Testing

The app has been taken to a number of events including the STEM for Girls event at the University of Sheffield in March 2020. On this occasion, it was used by approximately 60 students in Years 7–9 and they gave positive verbal feedback. The students at this event used an Arduino version of the game, with one joystick for speed and one for power.

The students arrived in small groups (4–6 per group) and were shown a range of components manufactured by both polymer and metal additive manufacturing, along with a desktop polymer 3D printer which was being used to make keyring tags. The printer was being monitored by a thermal camera, so they could see the live temperature of the polymer when it was extruded from the heating nozzle and as it cooled.

They were given an initial explanation of the additive manufacturing concepts, and were shown the app in ‘Baseline’ mode using the circle geometry, highlighting the areas which were too hot and too cold. They were then each given an opportunity to control their own build using the joysticks.

They generally found the first couple of hatches quite difficult, particularly those who opted to change both speed and power simultaneously, or where one student controlled speed and the other controlled power. Speed and power have opposing effects on the temperature, but they do not cancel out as they are differently weighted. Most then decided to concentrate on using just one of the joysticks, and were able to achieve good control for the first half of the build, while the hatch length was increasing.

The issue with the circle geometry is that after the halfway point the hatches start to get shorter again, requiring a reduction in energy to maintain constant temperature. The majority of students did not consider the geometry beforehand to identify a strategy, and so were taken surprise by this, but they quickly adjusted and were able to regain control.

Overall, most students achieved control for around 70% of the build duration, although some scored over 90%; as a comparison, the closed-loop control mode scores >98%. Only a couple of students struggled with the concept, and these were generally at the younger end of the cohort.

Each group was then shown the closed-loop control version, and was impressed by how quickly the temperature was brought within the acceptable range, and how well the algorithm responded to the changing geometry.

During discussions, it appeared that most students were aware of 3D printing with polymers, and were interested that it could be applied to metals. They particularly liked the articulated

components on display, and those with fine lattice details. They were keen to interact with the app and seemed comfortable with both the need to control temperature and the use of closed-loop control.

6. Code

The appendix includes pseudocode and the Matlab code for the key parts of the app. These should be sufficient to enable students to understand the logic behind the closed-loop control algorithm and create a comparable closed-loop control system of their own. Creating an identical version to the downloadable app will require the MATLAB App Designer, but similar control systems can be created in other software packages using the same approach.

Author contributions

Felicity Freeman developed the concept, created the app, wrote the manuscript and ran the app at the STEM for Girls event in March 2020. Lova Chechik carried out testing and ran the app at the Alloys for Additive Manufacturing Symposium in September 2019. All authors reviewed the manuscript.

Conflict of interest

The authors declare no competing financial interest.

Funding

MAPP: EPSRC Future Manufacturing Hub in Manufacture using Advanced Powder Processes (EP/P006566/1)

MIAMI: EPSRC Sheffield Impact Acceleration Account (EP/R511754/1)

Appendix: Code

This code is provided to enable students to see how the control algorithm identifies when the temperature is approaching the upper/lower limits, and how it then automatically adjusts the speed

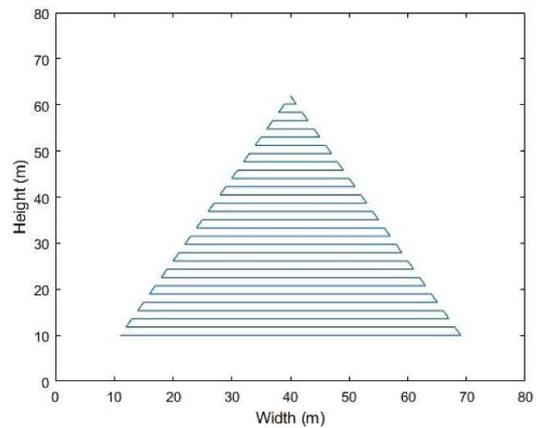


Figure A1. Example graph for triangle co-ordinates.

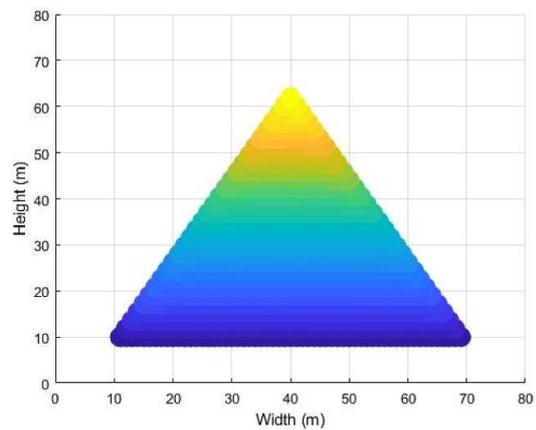


Figure A2. Example graph for triangle baseline temperatures.

and power to correct it. This can then be used as the basis for creating control algorithms for other processes.

The app has been written using MATLAB syntax, but a parallel version is provided in pseudocode to enable a comparable version to be created in alternative languages. In the pseudocode, the variables are identified in bold text. For additional support, please contact the corresponding author.

Appendix A. Baseline data

The example code below sets out how a set of baseline X/Y co-ordinates and corresponding temperatures can be created for the hatches in an

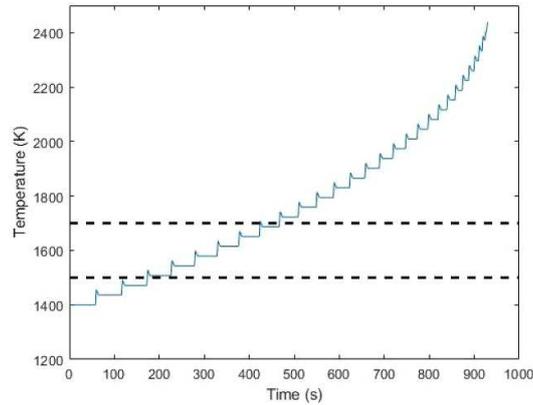


Figure A3. Example graph for triangle baseline time v. temperature.

```

%% Create X and Y co-ordinates of triangle hatches %%

Define variable Height for the total number of hatches in the triangle height
Set Height to 30
Define variable Width for total number of points across the base width
Set Width to ((Height x 2) + 1)

Define an empty list variable Y for all the Y coordinates

FOR every hatch from 1 (base of the triangle) to Height (top of the triangle):
Decrease the number of points in the hatch by 2 as the hatch number increases
Increase the value of the Y coordinate by 1 as the hatch number increases
Create new Y coordinates for each point across the hatch width
Append the new Y coordinates to the end of Y
END FOR

Define a list variable X for all the X co-ordinates
Set the first value of X to 1

FOR all values of Y from the second to the end:
IF Y is ODD
Add a new value to X by adding 1 to the previous value
ELSE (IF Y is EVEN)
Add a new value to X by subtracting 1 from the previous value
END IF
END FOR

Calculate the height for an equilateral triangle of base width defined in Width
Calculate the stretch factor necessary to scale the current Y range to an equilateral
triangle
Apply the stretch factor to all Y coordinates in Y

Add 10 to X (moves the data away from the origin for easier viewing)
Add 10 to Y (moves the data away from the origin for easier viewing)

Create a 2D line plot of X against Y
Set the axis limits to centre the data within the plot
Set the X axis label to ('Width (m)')
Set the Y axis label to ('Height (m)')

%%Create X and Y co-ordinates of triangle
Height=30;
Width=(Height*2)+1;

Y=[];

for hatchID=1:(Height)
    Y_add=(hatchID)*(ones((Width-(2*hatchID)),1));
    Y=cat(1,Y,Y_add);
end

X(1,1)=1;

for i=2:size(Y,1)
    if mod(Y(i,1),2)==1
        X(i,1)=X((i-1),1)+1;
    else
        X(i,1)=X((i-1),1)-1;
    end
end

Equilateral_Height=(Width-1)*((3^(1/2))./2);
Height_Factor=Equilateral_Height./(max(Y)-min(Y));
Y=((Y-1)*Height_Factor);

X=(X+10);
Y=(Y+10);

plot(X,Y)
axis([0 80 0 80]);
xlabel('Width (m)');
ylabel('Height (m)');
    
```

equilateral triangle. A similar approach can be used for other geometries.

The values for the starting temperature, hatch increase, turn increase and upper and

Beat the machine (learning): metal additive manufacturing and closed-loop control

```
%% Create baseline temperature – assumes fixed speed and power %%

Define variable Starting_Temp for the initial temperature and set to 1400
Define variable Hatch_Increase for the effect of shortening hatches and set to 20
Define variable Turn_Increase for the effect of turning the corner and set to 20
Define variable Upper_Limit for the upper temperature limit and set to 1700
Define variable Lower_Limit for the lower temperature limit and set to 1500

Define a list variable T
Set T equal to the (Starting_Temp + ((Y - 10) x Hatch_Increase))

FOR all values of Y from the second to (end minus 4)
  IF the Y coordinate is the same as the previous Y coordinate
    Do not change T
  ELSE (IF the Y coordinate is NOT the same as the previous Y coordinate)
    Add 100% of Turn_Increase to the corresponding value in list T
    Add 80% of Turn_Increase to the (corresponding + 2) value in list T
    Add 60% of Turn_Increase to the (corresponding + 3) value in list T
    Add 40% of Turn_Increase to the (corresponding + 4) value in list T
    Add 20% of Turn_Increase to the (corresponding + 5) value in list T
  END IF
END FOR

Create a list variable Z, the same length as Y, with every value equal to 1

Create a 3D scatter plot of X, Y, Z where the marker colour is defined by T
Set the view orthogonal to the X/Y plane
Set the axis limits to centre the data within the plot
Set the X axis label to ('Width (m)')
Set the Y axis label to ('Height (m)')
```

```
%%Create baseline temperatures – assumes fixed speed and power

Starting_Temp=1400;
Hatch_Increase=20;
Turn_Increase=20;
Upper_Limit=1700;
Lower_Limit=1500;

T=Starting_Temp+((Y-10)*Hatch_Increase);

for i=2:(size(Y,1)-4)
  if Y(i,1)==Y((i-1),1)
    T(i,1)=T(i-1,1);
  else
    T(i,1)=T(i,1)+Turn_Increase;
    T((i+1),1)=T((i+1),1)+(Turn_Increase*0.8);
    T((i+2),1)=T((i+2),1)+(Turn_Increase*0.6);
    T((i+3),1)=T((i+3),1)+(Turn_Increase*0.4);
    T((i+4),1)=T((i+4),1)+(Turn_Increase*0.2);
  end
end

Z=(Y./Y);
scatter3(X,Y,Z,200,T,'filled')
view(0,90)
axis([0 80 0 80]);
xlabel('Width (m)');
ylabel('Height (m)');
```

```
%% Create timesteps using a baseline speed %%

Create variable Speed and set to 1

Create list variable time and set the first value to 0
This will be the elapsed time to reach each coordinate

FOR all values of Y from the second to the end
  Calculate Distance between the current (X,Y) pair and the previous (X,Y) pair
  Use Speed to calculate dt, the time taken to travel Distance
  Calculate a new value of elapsed time by adding dt to the previous value
  Append the new value of elapsed time to the end of the time
END

Create a line plot of T against time
Set the axis limits to centre the data within the plot
Set the X axis label to ('Time (s)')
Set the Y axis label to ('Temperature (K)')
Add a horizontal line across the whole X axis extent at the Upper_Limit
Add a horizontal line across the whole X axis extent at the Lower_Limit
```

```
%%Create timesteps using a baseline speed

Speed=1;

time(1,1)=0;

for i=2:size(Y,1)
  XX=(X(i,1)-X((i-1),1))^2;
  YY=(Y(i,1)-Y((i-1),1))^2;
  distance=(XX+YY)^(0.5);
  dt=(distance./Speed);
  time(i,1)=time((i-1),1)+dt;
end

plot(time,T)
axis([0 1000 1200 2500])
xlabel('Time (s)');
ylabel('Temperature (K)');
line('XData', [0 1000], 'YData', [Upper_Limit Upper_Limit]);
line('XData', [0 1000], 'YData', [Lower_Limit Lower_Limit]);
```

```
%% Draw graphs using a FOR loop and a time delay to give 'live' updates %%

Create figure window
Keep figure window open while making changes
Create a 3D scatter plot using the only the first values of X, Y, Z, T
Set the view orthogonal to the X/Y plane
Set the axis limits to centre the data within the plot
Set the X axis label to ('Width (m)')
Set the Y axis label to ('Height (m)')

FOR all values of Y from the second to the end
  Pause by 0.001 seconds
  Update 3D scatter plot using data from 1 to the current value for X, Y, Z, T
  Redraw graph
END FOR
```

```
%%Draw graphs using a FOR loop and a time delay to give 'live' updates

figure(1);
hold on;
Graph=scatter3(X(1,:),Y(1,:),Z(1,:),200,T(1,:),'filled');
view(0,90)
axis([0 80 0 80]);
caxis([1200, 2500]);
xlabel('Width (m)');
ylabel('Height (m)');

for k = 2:(size(Y,1))
  pause(0.001);
  Graph=scatter3(X(1:k,:),Y(1:k,:),Z(1:k,:),200,T(1:k,:),'filled');
  drawnow;
end
```

lower limits can all be modified to provide different scenarios with varying levels of sensitivity.

Appendix B. Closed loop control

The equations governing the relationship between the power factor, speed factor and temperature can

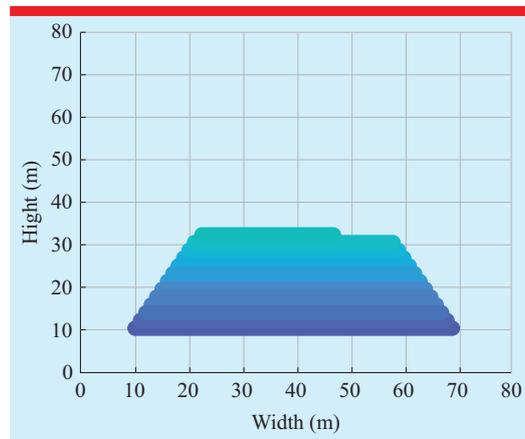


Figure A4. Example graph for 'live' updating.

```
%% Closed loop control %%
%% The magnitude of the changes in speed and power can be modified
%% The way that temperature is affected by speed and power can be modified
%% The selection of these will determine how successful the algorithm is for the application
%% Real applications require multiple iterations to identify a good control algorithm
%% Further refinements could include limits on the magnitude of power and speed
```

Create list variable **Power** and set the first value to 1, this will be the Power Factor
 Create list variable **Speed** and set the first value to 1, this will be the Speed Factor

Create a list variable **T_new** as a copy of **T**, this is to allow modification without overwriting

Create figure window
 Keep figure window open while making changes
 Create a 2D line plot of **T_new** against **time** using the only the first values
 Set the axis limits to centre the data within the plot
 Set the X axis label to ('Time (s)')
 Set the Y axis label to ('Temperature (K)')
 Add a horizontal line across the whole X axis extent at the **Upper_Limit**
 Add a horizontal line across the whole X axis extent at the **Lower_Limit**

```
FOR all values of Y from the second to the end
    IF T is less than (Lower_Limit + 5)
        Add a new value to Power by adding 0.01 to the previous value
        Add a new value to Speed by subtracting 0.01 from the previous value
    ELSE
        IF T is greater than (Upper_Limit - 5)
            Add a new value to Power by subtracting 0.01 from the previous value
            Add a new value to Speed by adding 0.01 to the previous value
        ELSE
            Add a new value to Power equal to the previous value
            Add a new value to Speed equal to the previous value
        END IF
    END IF

    Update T_new from the current value to the end by multiplying by the current value
    of Power ^ (1.3)
    Update T_new from the current value to the end by multiplying by the current value
    of Speed ^ (-0.5)

    Pause by 0.001 seconds
    Update 2D line plot using incremented slice from 1 to the current value
    Redraw graph
END FOR
```

Create a 3D scatter plot for **X, Y, Z, T_new**
 Create a 2D line plot for **Power** against **time**
 Create a 2D line plot for **Speed** against **time**

```
%%Closed loop control
%%The magnitude of the changes in speed and power can be modified
%%The way that temperature is affected by speed and power can be modified
%%The selection of these will determine how successful the algorithm is for the application
%%Real applications require multiple iterations to identify a good control algorithm
%%Further refinements could include limits on the magnitude of power and speed

Power(1,1)=1;
Speed(1,1)=1;

T_new=T;

figure(2);
hold on;
plot(time(1,:),T_new(1,:));
axis([0 1000 1200 2500]);
xlabel('Time (s)');
ylabel('Temperature (K)');
line('XData', [0 1000], 'YData', [Upper_Limit Upper_Limit]);
line('XData', [0 1000], 'YData', [Lower_Limit Lower_Limit]);

for k = 2:(size(Y,1))
    if T_new(k,1)<(Lower_Limit-5)
        Power(k,1)=Power(k-1,1)+0.01;
        Speed(k,1)=Speed(k-1,1)-0.01;
    else
        if T_new(k,1)>(Upper_Limit+5)
            Power(k,1)=Power(k-1,1)-0.01;
            Speed(k,1)=Speed(k-1,1)+0.01;
        else
            Power(k,1)=Power(k-1,1);
            Speed(k,1)=Speed(k-1,1);
        end
    end

    T_new(k:end)=T(k:end).*((Power(k,1))^(1.3)).*((Speed(k,1))^( -0.5));

    pause(0.001);
    plot(time(1:k,:),T_new(1:k,:));
    drawnow;
end

figure(3);
hold on;
scatter3(X,Y,Z,200,T_new,'filled')
view(0,90)
axis([0 80 0 80]);
xlabel('Width (m)');
ylabel('Height (m)');

figure(4);
hold on;
plot(time,(Power*100));
axis([0 1000 60 140]);
xlabel('Time (s)');
ylabel('Power Factor (%)');

figure(5);
hold on;
plot(time,(Speed*100));
axis([0 1000 60 140]);
xlabel('Time (s)');
ylabel('Speed Factor (%)');
```

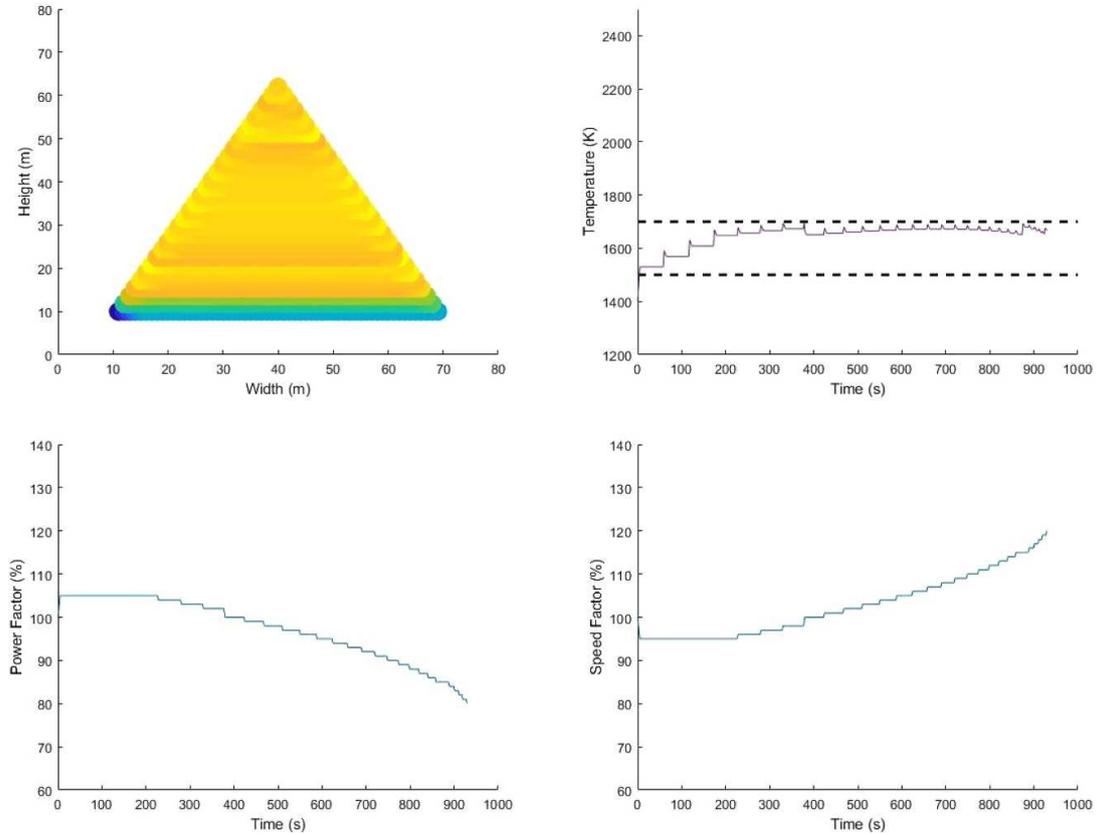


Figure B1. Example graphs for closed loop control.

all be modified to represent different scenarios. These specific equations are highlighted by red text/boxes.

Appendix C. User control

Running with user control requires a manual input, which could be from an Arduino, a keyboard or another external device. The code required to allow communication between that

device and the code will depend on the individual combination of device and language chosen. The approach is comparable with the Closed Loop Control code, except for some minor modifications.

As with the Closed Loop Control, the equations controlling how the user-controlled power and speed factors affect the temperature can be modified to give different levels of sensitivity. These are highlighted in the red text/boxes.

```
%% User control %%
%% The magnitude of the changes in speed and power can be modified
%% The way that temperature is affected by speed and power can be modified
%% The selection of these will determine how successful the algorithm is for the application
%% Real applications require multiple iterations to identify a good control algorithm
%% Further refinements could include limits on the magnitude of power and speed
```

Create list variable **Power** and set the first value to 1, this will be the Power Factor
 Create list variable **Speed** and set the first value to 1, this will be the Speed Factor

Create a list variable **T_new** as a copy of **T**, this is to allow modification without overwriting

Create figure window

Keep figure window open while making changes
 Create a 2D line plot of **T_new** against **time** using the only the first values
 Set the axis limits to centre the data within the plot
 Set the X axis label to ('Time (s)')
 Set the Y axis label to ('Temperature (K)')
 Add a horizontal line across the whole X axis extent at the **Upper_Limit**
 Add a horizontal line across the whole X axis extent at the **Lower_Limit**

FOR all values of **Y** from the second to the end
 Define a variable **UserPower** controlled by the external device
 Define a variable **UserSpeed** controlled by the external device

Set the current value of **Power** equal to the previous value multiplied by **UserPower**
 Set the current value of **Speed** equal to the previous value multiplied by **UserSpeed**

Update **T_new** from the current value to the end by multiplying by the current value of **Power** ^(1.3)
 Update **T_new** from the current value to the end by multiplying by the current value of **Speed** ^(-0.5)

Pause by 0.001 seconds
 Update 2D line plot using incremented slice from 1 to the current value
 Redraw graph

END FOR

Create a 3D scatter plot for **X, Y, Z, T_new**
 Create a 2D line plot for **Power** against **time**
 Create a 2D line plot for **Speed** against **time**

```
%%User control
%%The magnitude of the changes in speed and power can be modified
%%The way that temperature is affected by speed and power can be modified
%%The selection of these will determine how successful the algorithm is for the application
%%Real applications require multiple iterations to identify a good control algorithm
%%Further refinements could include limits on the magnitude of power and speed

Power(1,1)=1;
Speed(1,1)=1;

T_user=T;

figure(2);
hold on;
plot(time(1,:),T_user(1,:));
axis([0 1000 1200 2500]);
xlabel('Time (s)');
ylabel('Temperature (K)');
line('XData', [0 1000], 'YData', [Upper_Limit Upper_Limit]);
line('XData', [0 1000], 'YData', [Lower_Limit Lower_Limit]);

for k = 2:(size(Y,1))
    %*UserPower*
    %*UserSpeed*
    Power(k,1)=Power(k-1,1)*UserPower;
    Speed(k,1)=Speed(k-1,1)*UserSpeed;
    T_user(k:end)=T(k:end).*(Power(k,1)^(1.3)).*(Speed(k,1)^(-0.5));

    pause(0.001);
    plot(time(1:k,:),T_user(1:k,:));
    drawnow;
end
```

ORCID iDs

Felicity S H B Freeman 

<https://orcid.org/0000-0002-7402-1790>

Lova Chechik  <https://orcid.org/0000-0002-7626-2694>

Received 23 April 2020, in final form 22 May 2020

Accepted for publication 4 June 2020

<https://doi.org/10.1088/1361-6552/ab9957>

References

- [1] Wikipedia Alloy wheel (https://en.wikipedia.org/wiki/Alloy_wheel#Forging)
- [2] Wolfrace Wheels How alloy wheels are made (<https://www.wolfrace.com/how-wheels-are-made/>)
- [3] Gibson I, Rosen D W and Stucker B 2010 *Additive Manufacturing Technologies: 3D Printing, Rapid Prototyping, and Direct Digital Manufacturing* (New York: Springer) (<https://doi.org/10.1007/978-1-4419-1120-9>)
- [4] Frazier W E 2014 Metal additive manufacturing: a review *J. Mater. Eng. Perform.* **23** 1917–28
- [5] Gu D D, Meiners W, Wissenbach K and Poprawe R 2012 Laser additive manufacturing of metallic components: materials, processes and mechanisms *Int. Mater. Rev.* **57** 133–64
- [6] Pham M-S, Liu C, Todd I and Lertthanasarn J 2019 Damage-tolerant architected materials inspired by crystal microstructure *Nature* **565** 305–11
- [7] Arcam AB 2018 EBM for orthopedic implants (www.arcam.com/solutions/orthopedic-implants/)
- [8] Liu S and Shin Y C 2019 Additive manufacturing of Ti6Al4V alloy: a review *Mater. Des.* **164** 107552
- [9] CFM International Jet Engines The LEAP Engine (<https://www.cfmaeroengines.com/engines/leap/>)
- [10] van Grunsven W, Hernandez-Nava E, Reilly G and Fabrication G R 2014 Mechanical Characterisation of Titanium Lattices with Graded Porosity *Metals (Basel)* **4** 401–9
- [11] Tammis-Williams S and Todd I 2017 Design for additive manufacturing with site-specific properties in metals and alloys *Scr. Mater.* **135** 105–10
- [12] General Electric What is additive manufacturing? GE additive (<https://www.ge.com/additive/additive-manufacturing>)
- [13] ASTM International 2016 *ISO/ASTM 52900:2015(E). Standard Terminology for Additive Manufacturing—General Principles—Terminology* vol 52900

Beat the machine (learning): metal additive manufacturing and closed-loop control

- (<https://compass.astm.org/download/ISOASTM52900.28913.pdf>)
- [14] Barak S, Mudgil D and Khatkar B S 2014 Effect of flour particle size and damaged starch on the quality of cookies *J. Food Sci. Technol.* **51** 1342–8
- [15] King W E, Barth H D, Castillo V M, Gallegos G F, Gibbs J W, Hahn D E, Kamath C and Rubenchik A M 2014 Observation of keyhole-mode laser melting in laser powder-bed fusion additive manufacturing *J. Mater. Process. Technol.* **214** 2915–25
- [16] Darvish K, Chen Z W and Pasang T 2016 Reducing lack of fusion during selective laser melting of CoCrMo alloy: effect of laser power on geometrical features of tracks *Mater. Des.* **112** 357–66
- [17] Kwon O, Kim H G, Ham M J, Kim W, Kim G-H, Cho J-H, Kim N I and Kim K 2020 A deep neural network for classification of melt-pool images in metal additive manufacturing *J. Intell. Manuf.* **31** 375–86
- [18] Mathworks Matlab runtime (<https://uk.mathworks.com/products/compiler/matlab-runtime.html>)



Felicity Freeman is a post-doctoral researcher at the University of Sheffield, currently working on feedstock control for direct energy deposition. She completed her PhD in 2019, where she investigated how the build parameters can be exploited to deliver microstructural control in laser powder-bed fusion. She specifically focussed on the manufacture of magnetically graded components,

achieved by spatial control of microstructure within a single composition of stainless steel. Before moving into academia, Felicity spent over a decade in industry, specialising in non-contact metrology for single-crystal turbine blade castings.



Lova Chechik is a PhD student at the University of Sheffield, UK. His research is focussed on thermal monitoring and control in Metal Additive Manufacturing.



Iain Todd is based at the University of Sheffield where he directs the EPSRC Future Manufacturing Hub In Manufacture using Advanced Powder Processes (www.MAPP.ac.uk). He presently holds the RAE/ GKN Aerospace Research chair in Additive Manufacture and Advanced Structural Metallics. His work lies at the interface between manufacturing technology and materials processing science and concerns the development of understanding of underlying physical principles related to materials processing to better control material form, integrity and function and hence economic value. He holds a BEng and Phd from the University of Sheffield.