MACHINE LEARNING ENABLED POWDER SPREADING PROCESS MAP FOR METAL ADDITIVE MANUFACTURING (AM)

Wentai Zhang[1], Akash Mehta[1], Prathamesh S. Desai[1], Prof. C. Fred Higgs III[1,2]

[1] Carnegie Mellon University, Pittsburgh, PA 15213
[2] Rice University, Houston, TX 77005

Abstract

The metal powder-bed AM process involves two main steps: the spreading of powder layer and selective fusing or binding the spread layer. Most AM research is focused on powder fusion. Powder spreading is more rarely studied but is of significant importance for considering the quality of the final part and total build time. It is thus essential to understand how to modify the spread parameters such as spreader speed, to generate layers with desirable roughness and porosity. A computational modeling framework employing Discrete Element Method (DEM) is applied to simulate the spreading process, which is difficult to study experimentally, of Ti-6Al-4V powder onto smooth substrates. Since the DEM simulations are computationally expensive, machine learning was employed to interpolate between the highly non-linear results obtained by the running a few DEM simulations. Eventually, a spreading process map is generated to determine which spreader parameters can achieve the desired surface roughness and spread speed. This eventually saves the total time for printing and reduces the cost of build.

Keywords: Additive Manufacturing (AM), powder spreading process map, Discrete Element Method, Machine Learning

Nomenclature

Symbol	Meaning
Δ	Overlap of a particle with another particle or geometry
ϕ	Diameter of a spherical particle
K	Stiffness of spring in a spring-dashpot system
β	Damping of dashpot in a spring-dashpot system
n, t	Subscripts: normal and tangential directions respectively
ε	Coefficient of restitution
m	Mass
V	Speed
μ	Coefficient of sliding friction
$\underline{e_t}$	Unit vector along the tangential direction
U	Translation speed of the spreader

ω	Rotational speed of the spreader
Vs	Volume of powder spread per unit time per unit width of spreader
Rq	Roughness of spread layer or substrate
L	Loss function
N	Total number of training samples
Y, O	Actual and target output vectors respectively
α	Learning rate
λ	Regularization parameter
R	Correlation coefficient

Introduction

Powder-bed additive manufacturing (AM), colloquially known as three-dimensional (3D) printing, is one of the few types of technologies slated to disrupt the traditional manufacturing industry predominantly dependent on casting, molding and subtractive manufacturing. The state-of-the-art powder-bed 3D printers are optimized to work only with a handful of powders and the parts built using such printers have rough exterior and porous interior. The 3D printing process used involves repetitive spreading of powder and selective fusing or binding of particles in the spread layer until the entire geometry is 3D printed (Strondl A. et al., 2015) refer **Fig. 1**. A commonly used metal powder made of Ti-6Al-4V and a cylinder printed using this powder are shown in **Fig. 2a** and **2b** respectively. An optical scan of the top surface of printed cylinder can be seen to have noticeable striations, see **Fig. 2c,** which make the part non-isotropic with unpredictable mechanical properties and rough exterior. Most of the existing AM research is clustered around fusing (e.g., laser sintering or melting) process optimization (Beuth et al. 2013; Gockel et al. 2014). The step of powder spreading is rarely studied and makes use of machine default spread settings; however uniform spreading of powder layer is mandatory to 3D print dense and isotropic parts with a smooth surface finish. Only a handful of studies (Herbold et al. 2015; Haeri et al. 2016; Parteli & Pöschel 2016; Mindt et al. 2016) have attempted to answer the influence of spreading step in the entire 3D printing process.

Herbold et al. in 2015 performed a computational study which used 40-particles square by 10-particles deep domain with particle sizes as seen in real AM metal powders but no justification was provided as to the choice of the domain size and no experimental validation was provided for the spreading simulations. Parteli & Pöschel in 2016 incorporated complex shapes of powder particles and provided relationships between spreader speeds and layer roughness. Using only model simulation results, in this work they employed a small domain size with periodic boundary conditions. Haeri et al. in 2016 and Mindt et al. in 2016 followed a small domain simulation approach as done by Parteli & Pöschel in 2016. Haeri et al. in 2016 made use of rod-like particles and two spreader geometries, a blade and a roller, comprised of spherical particles, thereby adding an unreal roughness to the spreaders. The study conducted by Mindt et al. in 2016 has accounted for true geometry of the previously printed layer along with particle size distribution. They have also simulated the fusion process. Similar to the aforementioned works, the domain simulated in their work was smaller than the real size of a build platform.

The authors of this paper aim to study the *spreadability* of AM powders, i.e., the ability to spread or make powders flow under a given compressive load, by following a synergistic approach

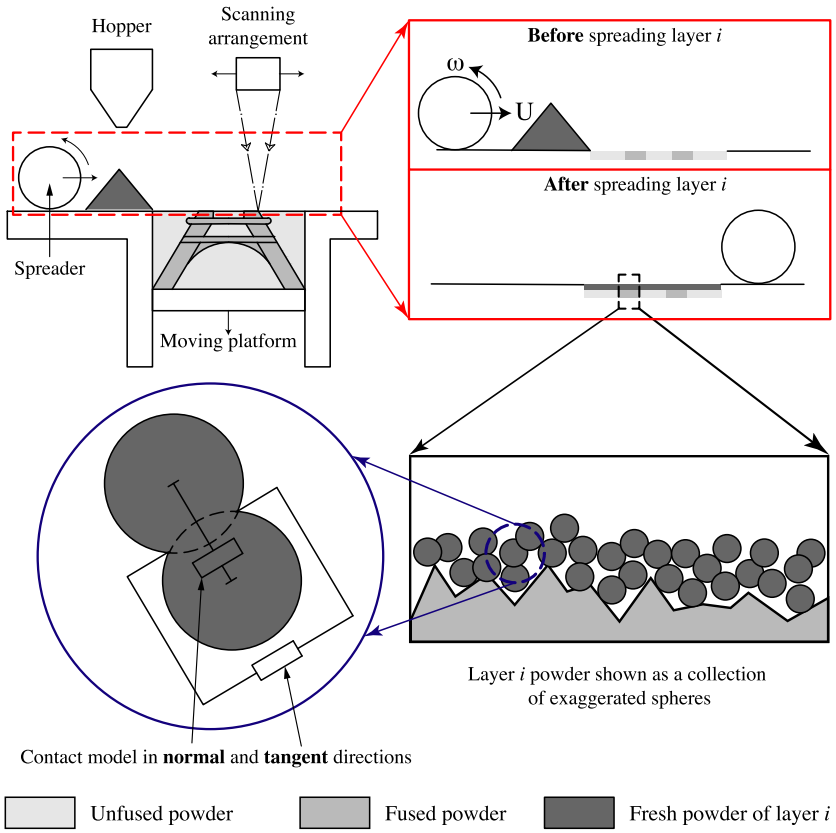
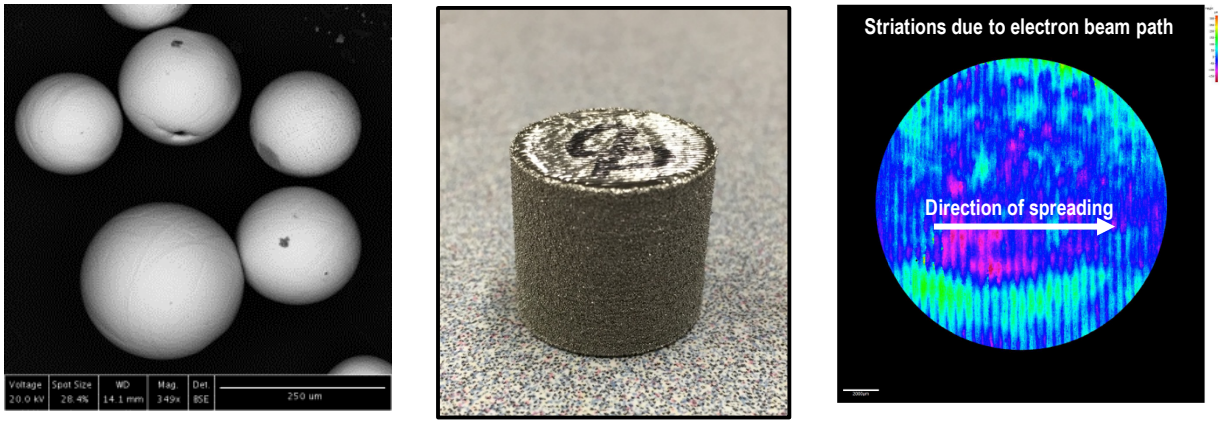


Figure 1 Schematic of powder-bed AM process (top left) with insets showing powder spreading (top right and bottom right) and contact model used in physics-based modeling (bottom left)



(a) **(b)** **(c)**

Figure 2 (a) SEM micrograph of 250μm Ti-6Al-4V powder, **(b)** 3D printed cylinder from an AM machine using electron beams for binding the metal powder shown in (a), **(c)** Optical image of the top surface of (b)

involving interplay of experiments, physics-based modeling and machine learning as summarized in Section II. Section III describes the *in silico* virtual spreading experiments performed in scenarios and at scales similar to those found in real 3D printers using physics-based GPU-

optimized Discrete Element Method (DEM). Since the DEM simulations are computationally expensive, only a few such simulations are run following a Design of Simulations (DoS) approach. Subsequently, in Section IV, machine learning has been employed to interpolate between the highly non-linear results obtained by the DEM simulations. Spreading process maps generated using such a synergistic approach can be used to find the most efficient spreading parameters to achieve a desirable surface finish.

Methodology

The problem of studying the spreadability of AM powders is twofold, firstly, it is difficult to study this problem experimentally inside a real 3D printer, due to the difficulty involved in characterizing the spread layer parameters without interfering with the environmental conditions required for working with Ti-6Al-4V powder. The safety issues associated with the handling of AM powders such as toxicity, flammability and explosivity make a trial-and-error approach, common with experimental studies, unrealistic and unsafe (Huang et al. 2013). This first problem makes the experimental study not only difficult but also expensive. Secondly, computational study of this problem is also not trivial as the DEM, most well suited among other computational techniques, is based on Lagrangian principles and has no simple constitutive laws for AM powders (Bharadwaj 2012). Therefore, a synergistic, three-step approach as shown in **Fig. 3** is used to predict spreadabilty of AM powders. The first step involves the characterization of the AM powder using a powder rheometer (Dougherty 2016) and using the data for calibrating a 'virtual powder' which behaves similar to the real AM powder as discussed in previous works of authors. This rheometer also serves the purpose of experimentally validating the DEM model and exposes the powder to loadings similar to those seen in powder spreading. The second step involves the spreading simulation study of this rheometry-validated virtual powder and comparison to the real spreading, if possible but has not been done in this study. Finally, referring to **Fig. 3**, the simulation data is used to train and test regression algorithms based on machine learning, e.g., back propagation neural networks, to generate spreading process maps. These maps show the relations between 3D printer operator's input parameters e.g., spreader speeds and spread layer parameters.

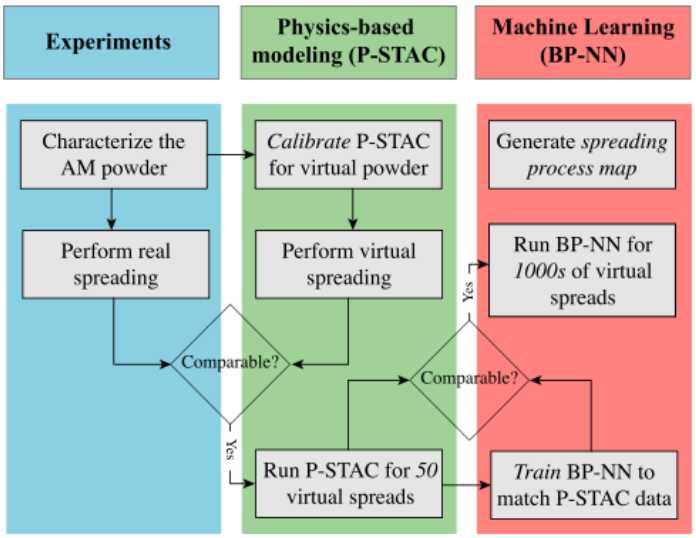


Figure 3 Synergy between experiments, physics-based DEM and machine learning

The following sections describe the second and third steps of this methodology and discuss the results obtained using these steps.

Physics-based DEM Modeling

The Discrete Element Method (DEM) is used in this study to simulate the powder spreading process in AM. In this study, DEM makes use of uniformly sized, 235,000 smooth spherical, cohesionless elements of 250μm diameter to represent the AM powder. **Figure 2a** shows an SEM image of a Ti-6Al-4V powder, commonly used in AM, which has the maximum size of about 250μm. Also shown are a 3D printed cylinder and its top-surface optical scan in **Fig. 2b** and **Fig. 2c** respectively. The striations seen in **Fig. 2c** are in the direction of spreading and can be attributed to uneven heating of the spread layer by the electron beams (Ho et al. 2007). The roughness *Rq* of the top surface of this 3D printed cylinder was about 46.5μm. However, this study is carried out on an ideally smooth substare. The length scales involved in these simulations, which have particles with sizes of 10 to a few hundred micrometers and the spread layer of 10's of centimeters in size, drastically increase the number of computations as the particle count can easily reach millions. In order to simulate this problem in realistic times without compromising the accuracy of the simulation, the DEM code is parallelized to run on a Graphics Processing Unit (GPU). There are two different types of collisions involved in the simulation of powder spreading in AM, namely powder particles colliding with other powder particles and powder particles colliding with the solid surfaces of the spreader. Each type of collision has its own computational challenges. The former particle-particle collision requires an efficient neighborhood search (Ferrez 2001) while the latter, particle-surface collision, requires an accurate representation of the surface geometry. The neighborhood search is the most time-consuming step in a DEM simulation. Hence, a verlet-based (Ferrez 2001; Mpagazehe 2013; NVIDIA 2008) efficient neighborhood search algorithm is employed using a technique called 'spatial binning' (Green 2013) to further improve the performance of the solver.

Contact Model

The contact model used in this study is comprised of two damped Hookean-springs (Bharadwaj 2012), one in normal (subscript *n*) and other in shear or tangential direction (subscript *t*), as shown in Fig. (2). The K and β stand for stiffness and damping respectively and the expressions for these are as given by equations 1, 2 (Mishra & Murty 2001):

$$K_n = \frac{f^2 m_{eq} V_{max}^2}{\phi^2} \; ; f = \frac{\phi}{\Delta_{max}} \qquad (1) \qquad \beta_n = -2\ln(\varepsilon)\left[\frac{K_n m_{eq}}{\pi^2 + \{\ln(\varepsilon)\}^2}\right]^{1/2} \qquad (2)$$

Here m_{eq} stands for the equivalent mass of colliding particles, having diameter ϕ and constant coefficient of restitution ε which is independent of impact velocity (Bharadwaj 2012). This m_{eq} is one half of the harmonic mean of the individual masses. V_{max} and Δ_{max} are the estimated maximum speed and inter-particle penetration respectively for the simulation at hand. These values are usually guessed. A slider is also present in the shear direction. It limits the maximum frictional force in this direction, the value of which is equal to the product of sliding friction coefficient μ and normal reaction force $\underline{F_n}$ (given by Eq. 3). It is assumed that all the interactions cause particles to slide thereby nullifying the tangential damped Hookean spring. In other words, only the slider

acts in the shear direction. Therefore, the forces along the normal ($\underline{F_n}$) and tangential ($\underline{F_t}$) directions experienced by a colliding particle with an overlap of Δ with other particle or solid surface geometry, relative approach speed of $\dot{\Delta}$ and unit vector in shear direction as $\underline{e_t}$ can be represented as:

$$\underline{F_n} = K_n \underline{\Delta_n} - \beta_n \underline{\dot{\Delta}_n} \quad (3) \qquad \underline{F_t} = -\mu \left|\underline{F_n}\right| \underline{e_t} \quad (4)$$

This, in entirety, forms the DEM module of an in-house multi-purpose, multi-physics software called as the **Particle-Surface Tribology Analysis Code (P-STAC)**. P-STAC is used to perform spreading simulations as described below.

Design of Simulations (DoS) for virtual spreading

Spreading simulations require a set of contact force parameters which can make the virtual powder bulk behave in ways similar to a real AM powder. The density of these spherical particles is 4430kg/m^3 and is equal to the real AM Ti-6Al-4V powder shown in **Fig. 2a**. This virtual bulk is developed by using the calibration process mentioned in earlier works by authors of this study. The DEM parameters used in this study are summarized in **Table 1**. GPU parallelized P-STAC, though much faster than a CPU serial code, requires significant computational time to simulate a spread of about 235,000 powder particles. So only 45 simulations have been conducted using an n-factorial design of simulations (DoS) approach on the lines of design of experiments approach carried out by Asadi-Eydivand et al. (2016). The different parameters for spreading simulations, involving a roller as a spreader, are summarized in **Table 2**. The substrate is assumed to be perfectly smooth. These ranges in spreader speeds nicely cover the speeds seen on a real 3D printer. Sample simulation snapshots are shown in **Fig. 4**.

Table 1: DEM parameters used in spreading simulations

Property	Ti-6Al-4V powder interacting with		
	spreader	3D printed substrate	Ti-6Al-4V powder
ε	0.8$^{\#}$	0.8$^{\#}$	0.8$^{\#}$
μ	0.12$^{@}$	0.25$^{\#}$	0.185*

Note: * => value tuned via the DEM calibration process,
$^{@}$ => value measured using rheometer,
$^{\#}$ => assumed value

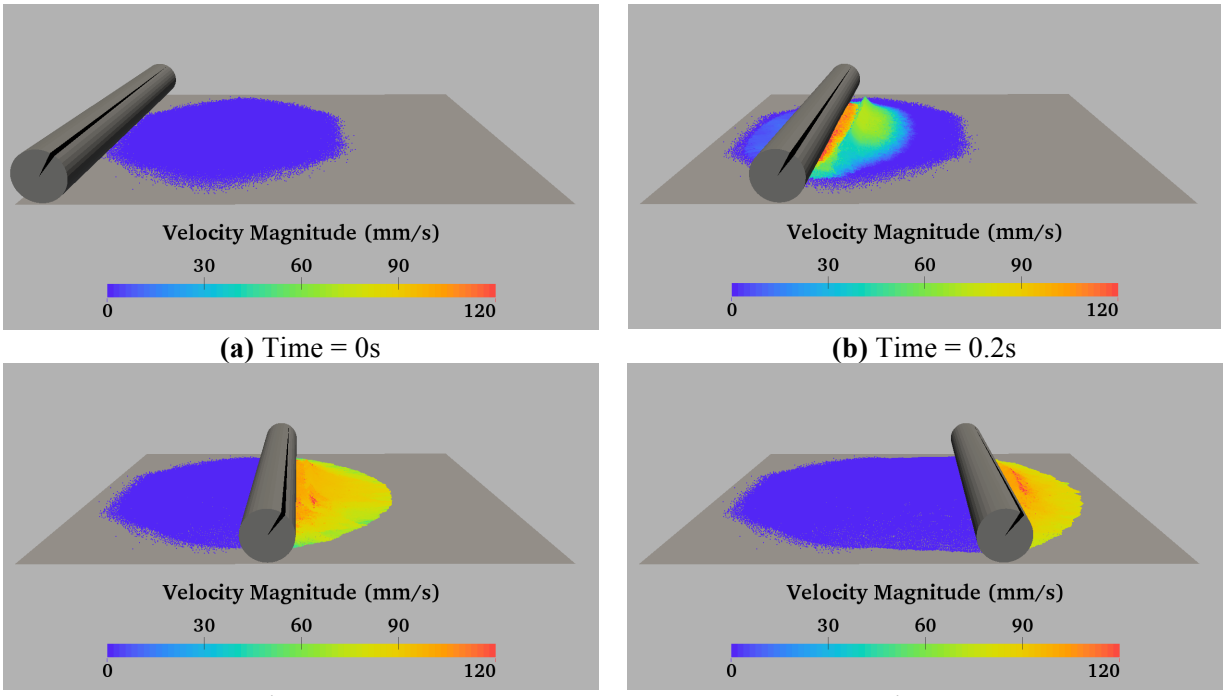
Table 2: Design of Simulations (DoS) for virtual spreading

Parameter	Value(s)
Spreader diameter (mm)	10
Spreader length (mm)	70
Spreader translation speed, U (mm/s)	40, 55, 70, 85, 100
Spreader rotation speed, ω (rad/s)	0, 5, 10, 15, 20, -5, -10, -15, -20

Figure 4 shows spreading of virtual 250µm Ti-6Al-4V powder over a flat substrate at spreader translation speed of 100mm/s and no rotation. A black slit is shown on the spreader to visualize the direction of rotation, absent in this case, of the spreader. After the spreading simulation, the spread layer is critiqued for layer parameters as described below. On an average, each spreading simulation took 45 minutes to run on an nVIDIA GTX 1070 GPU.

Spread layer characterization

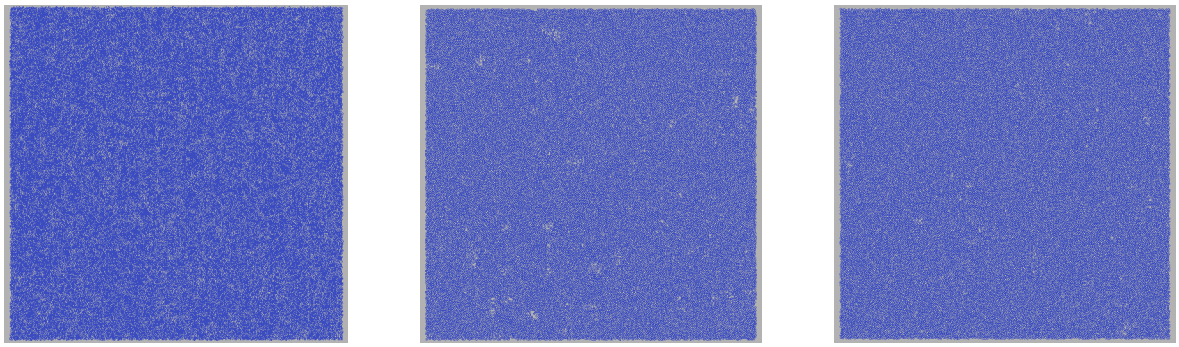
A 50mm x 50mm region centrally located above the substrate, after the spreading simulation has completed, is sampled for two important properties: volume of powder spread per unit time per unit width of the spreader, *Vs* and the roughness of the spread layer, *Rq*. *Vs* is indicative of the efficiency of the spreading while *Rq* is indicative of the qualitative aspect of the layer. The optimum values for *Vs* and *Rq* depend on the AM application. To calculate *Vs*, the mean height of the spread layer in the sampling region is multiplied by the spreader translation speed *U*. *Rq* is the standard deviation of the heights occupied by the spread layer in the sampling region. The sampling region for the case shown in **Fig. 4** is shown in **Fig. 5b** along with sampling regions for cases involving spreader translation speed U of 100mm/s and rotation speed ω of -20 rad/s (**Fig. 5a**) and spreader translation speed of 100mm/s and rotation speed of 20 rad/s (**Fig. 5c**). These spread layers over a flat substrate can be seen to have voids which result in porosity in the 3D printed part and can eventually cause failure of the part during loading due to stress concentrations.



(a) Time = 0s **(b)** Time = 0.2s

(c) Time = 0.4s **(d)** Time = 0.6s

Figure 4 Simulation snapshots of virtual spreading with roller having U = 100mm/s and ω = 0 on a flat substrate. Particles are colored by values of their velocity magnitude.



(a) **(b)** **(c)**

Figure 5 Sampling region of the spread layer: **(a)** U = 100mm/s and ω = -20rad/s, (**b)** U = 100mm/s and ω = 0rad/s, **(c)** U = 100mm/s and ω = 20rad/s

Machine-learning-based Spreading Predictions

The physics-based simulation results, as seen in the previous section, are highly non-linear and the simulation time, per spreading simulation, is quite high to perform a parametric study covering the entire range of spreader translation and rotation speeds, thereby resulting in a better understanding of the effect of these speeds on the spread layer parameters *Vs* and *Rq*. This problem is well suited to be solved using machine learning techniques to regress between the data obtained via design of spreading simulations from the previous section. Authors of this paper have chosen neural network to perform the regression over the datasets since neural networks (Basheer et al. 2000) can generate an unbiased fit over a dataset than other regression techniques which require assumptions about the function of the surface to be regressed over the dataset (Asadi-Eydivand et al. 2016; Chen et al. 2015; Jiang et al. 2014; Tourloukis et al. 2016;).

Back Propagation Neural Network (BP-NN) (Bishop 2006)

A neural network is a mathematical model of a biological neuron. In biological neurons, the dendrite receives electrical signals from the axons of other neurons; in the artificial neural network these electrical signals are represented as numerical values (Basheer et al. 2000). Generally, there are three kinds of layers in a neural network, namely the input layer, hidden layer(s) and output layer, see **Fig. 6**. The input layer is a vector of values which are given as conditions in the problem. Similarly, the output layer is also a vector of values which are the target solutions for the problem. In the case of studying the effect of spreader speeds on the spread layer, the input layer vector is spreader translation speed U and spreader rotation speed ω and the output layer vector is made of spread layer parameters *Vs* and *Rq*, as defined in the previous section. There may be a single hidden layer or multiple hidden layers in the network based on how the constructor defines the network. For this study, the neural network comprises of a single hidden layer. Within each hidden layer, a vector of values is calculated using the data from the previous layer and these values are generated by the network to represent some feature of the data. Each layer is connected with the next layer using weights. These weights form a matrix of linear factors. The product of the vector from a certain layer and the weights matrix is the vector of the next layer. This means that each node in the next layer is a linear combination of nodes from the previous layer. However, this network has only linear functions. Many real problems often have complex nonlinear

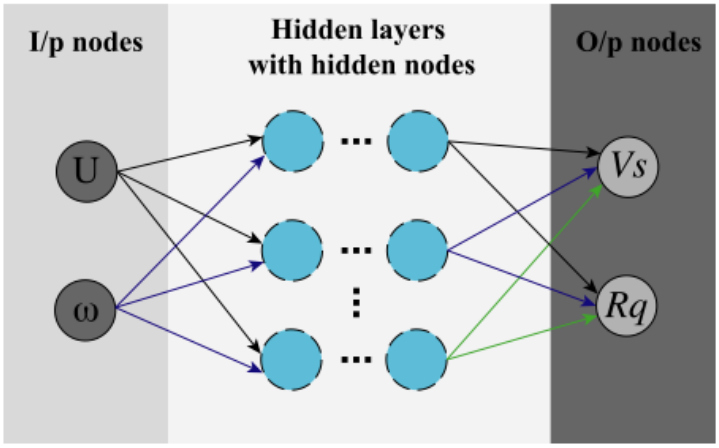


Figure 6 Schematic of a general neural network (NN) with multiple hidden layers with each hidden layer having multiple hidden nodes

relationships between input and output. So, a nonlinear activation function is commonly used to make the network nonlinear and allow for the learning of rather complicated problems.

In the present study, a sigmoid function, defined as $f(x) = \frac{1}{1+e^{-x}}$ or $f(x) = tan^{-1}(x)$, is used as an activation function. As the structure of the neural network has been defined, a useful way to train the network is back propagation (BP). In this training method, the target is the loss function which is commonly written as:

$$L = \frac{1}{N}\sum_{i=1}^{N} \|Y_i - O_i\|^2 \tag{5}$$

where N is the total number of training data. Y_i is the actual output vector for the i^{th} training data. O_i is the target output vector for the i^{th} training data. The loss function is implemented to find the difference between the real output and the target output. Therefore the training process is actually finding the minimum of the loss function. Here, gradient descent algorithm is implemented to minimize the loss function. The loss function can be regarded as a complex nonlinear function. A random initial point can be defined and the direction where the function has the fastest decreasing speed can be found by calculating the derivative on that point. Then a step along the function is taken with a fixed step size and the new point is acquired. Iterating many times in this fashion, the point will get closer and closer to the minimum point.

For the training of the neural network, first, the weights are randomly generated. Then outputs are calculated from the inputs and the random weights. Finally, the loss function value can be obtained and used as the updates for the weight:

$$W^{(n+1)} = W^{(n)} - \alpha \Delta W \tag{6}$$

where

$$\Delta W = \frac{\partial L}{\partial W} \tag{7}$$

In the above equations, α is the learning rate which will control the step size of gradient descent in each iteration. If α is too small, it may take a large number of iterations for the loss function to come to convergence. However, if α is too large the learning process may crash when the network is training. Oscillations will occur on the loss function value for each iteration.

Another main challenge of training the network is overfitting where the training error is decreasing but test error is increasing. Usually the reason is that the complexity of the network becomes much higher than the data itself and the weights have large magnitude. Hence L-2 regularization is implemented in the loss function to avoid overfitting:

$$L = \frac{1}{N}\sum_{i=1}^{N} \|Y_i - O_i\|^2 + \lambda \|W\|^2 \tag{8}$$

In the new loss function, the norm of all the weight is put into the loss function and λ is the parameter to control the level of regularization. In this way, as the loss function decreases, the magnitude of all the weights is secured to be small.

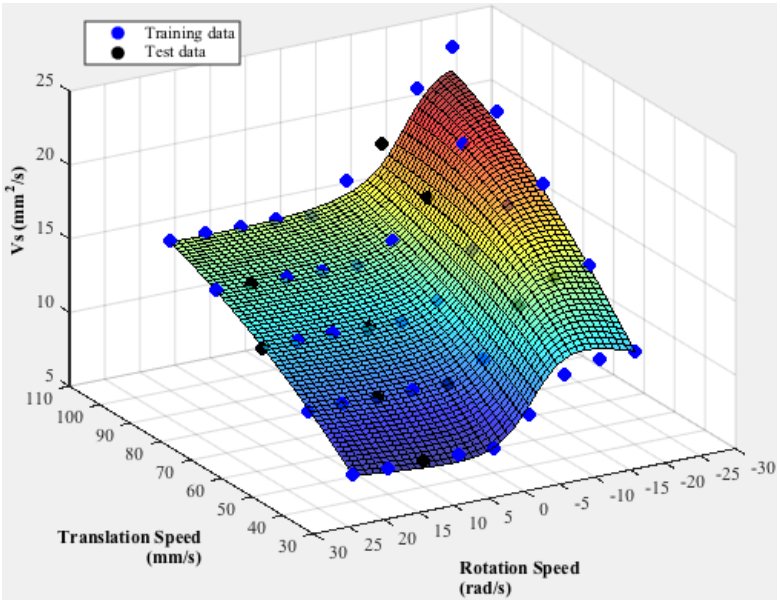
The parameters used for back propagation neural network (BP-NN) used to regress between the spreader speeds and spread layer parameters are listed in **Table 2**. Number of hidden nodes was decided by conducting a parametric study involving BP-NNs with increasing number of hidden nodes and 200 was chosen as a tradeoff between accuracy and computational efficiency. The learning rate and L2-regularization parameter were also decided by conducting numerical experiments but for the brevity of the paper these analyses are not presented. Regression results from the final BP-NN with parameters listed in **Table 3** are shown in **Fig. 7** and **Fig. 8** for spreading simulations over a flat substrate. The surfaces predicted by this BP-NN, refer **Fig. 7a** and **Fig. 8a**, nicely blanket the simulation data points, both training and test data points, generated via the Design of Simulations. Also shown is the correlation coefficient *R* between simulation results and predicted results in **Fig. 7b, 7c** and **Fig. 8b, 8c**. The near unity value of R for both training and test data points for each of the layer parameters *Vs* and *Rq*, suggests a near perfect regression. The error values and *R* values for the three substrates are listed in **Table IV**. BP-NN trained for spreading of Ti-6Al-4V powder on flat substrate is able to predict results with at least 97.5% accuracy. The normalized RMS error reported in the **Table IV** refers to the root mean square of the error in predicted spread layer parameters by BP-NN with respect to the ground truth generated using P-STAC and normalized by the range of the corresponding value.

Table 3: BP-NN parameters for spreading simulations over a particular substrate

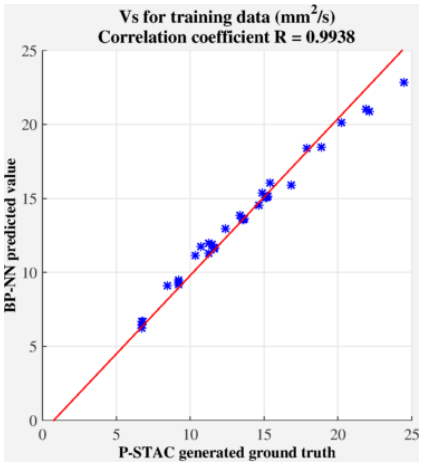
Parameter	Value	Parameter	Value
Number of training samples	35	Activation function for hidden layer	Sigmoid
Number of test samples	10	Activation function for output layer	Linear
Number of hidden layers	1	Learning rate α	0.0001
Number of hidden nodes	200	L2-regularization parameter λ	0.1

Table 4: BP-NN performance matrix

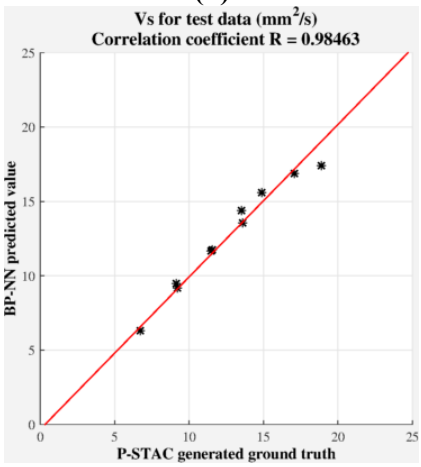
Vs				
Substrate *Rq* (μm)	**Normalized RMS error (%)**		**Correlation coefficient**	
	For training sample	**For test samples**	**For training samples**	**For test samples**
0	1.74	2.21	0.9938	0.98463
Rq				
Substrate *Rq* (μm)	**Normalized RMS error (%)**		**Correlation coefficient**	
	For training sample	**For test samples**	**For training samples**	**For test samples**
0	1.99	2.27	0.98264	0.98175



(a)



(b)



(c)

Figure 7 BP-NN regressed surface for *Vs*

Spreading process map

A spreading process map is shown in **Fig. 9**. This process map relates the 3D printer spreader parameters of translation U and rotational ω speeds to the spread layer parameters of *Vs* and *Rq*. The *Rq* of the spread layer increases as the rotation of spreader changes from anticlockwise (+) to clockwise (-) direction. This is due to the clockwise spreader rotation forcing spread of multiple layers as opposed to only one to two layers in the cases of no and anticlockwise rotational motion. For a constant rotational speed ω, the efficiency of spread increases at the translational speed increases. Conversely, the most efficient way to spread a layer, which is indicated by a larger *Vs*, of known roughness is to obtain the rightmost U-ω pair on the process map.

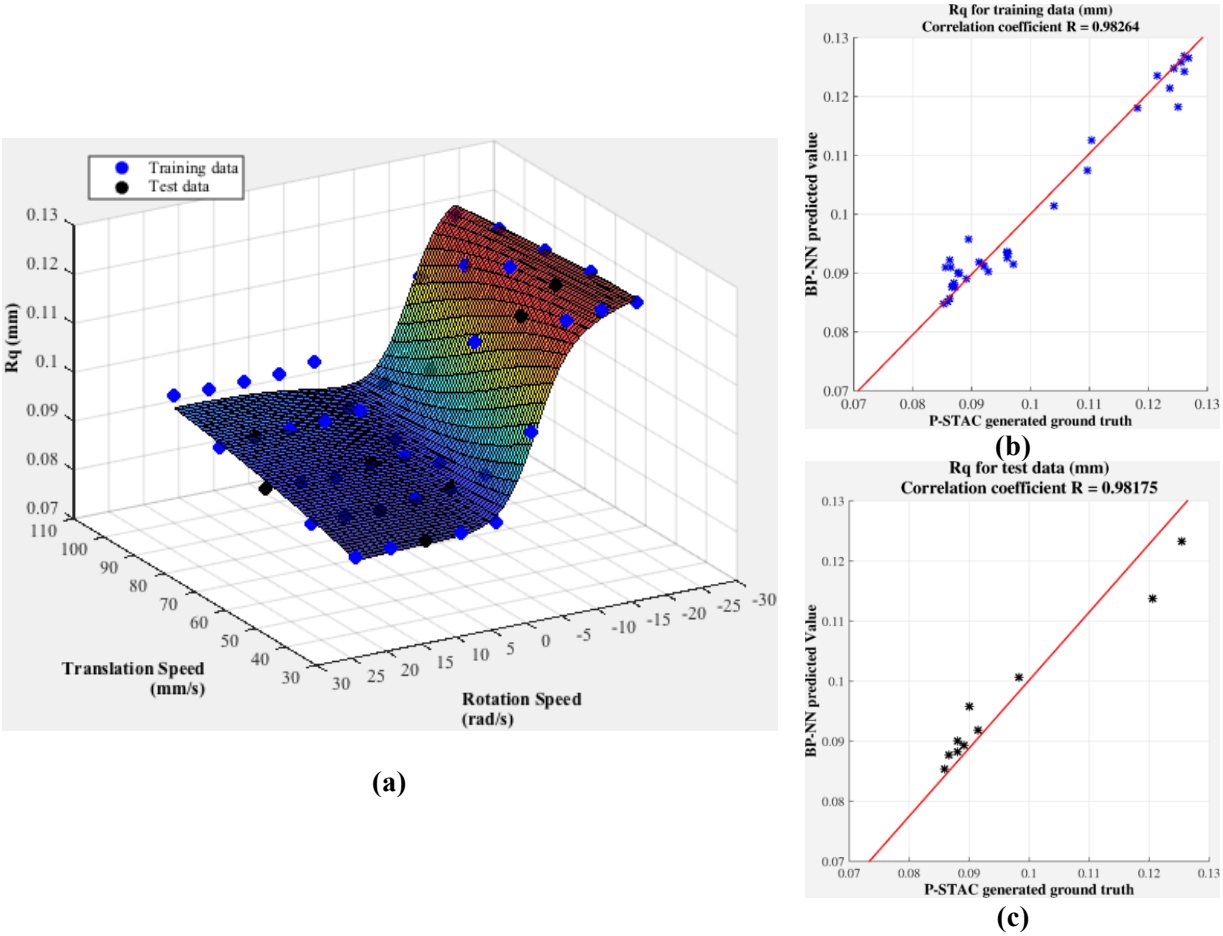


Figure 8 BP-NN regressed surface for *Rq*

Conclusions

To conclude, a synergistic framework based on DEM modeling and machine learning is presented to provide spreading predictions in powder-bed AM techniques. DEM was successfully applied to simulate spreading of virtual Ti-6Al-4V powder, which is modeled as 235,000 smooth spherical, cohesionless particles, on a smooth substrate. Back propagation neural network is used to regress between the highly non-linear spread layer parameters as a function of spreader speeds.

The spreading process maps thus developed can be made available to a 3D printer operator to efficiently print parts at desired roughness values within the intervals of contours on spreading process maps. These maps can also serve to aid with other factors such as thermal performance. For example, these maps can be used to introduce roughness in the spread layer to offset the uneven temperature distribution which is often encountered in the electron beam or laser based binding processes, thereby resulting in a smooth and uniform build or 3D printed layer. Further investigations will involve the incorporation of surface roughness of the preceding spread layers.

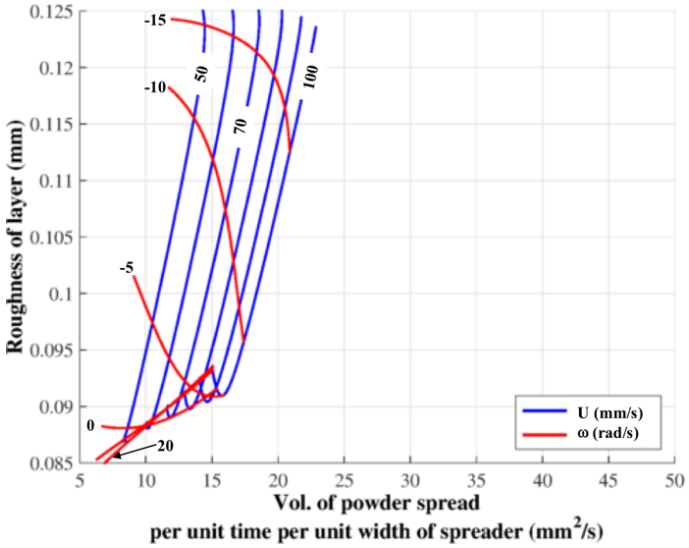


Figure 9 Spreading process map relating the Vs and Rq of the spread layer to the translation speed U and rotation speed ω of the spreader

Acknowledgements

Authors of this work would like to acknowledge SEM imaging of Ti-6Al-4V powder from Ross Cunningham and Anthony Rollett at Carnegie Mellon University (CMU). We also thank Luke Schime and Jack Beuth for the 3D printed cylinder. Finally, we thank Recep Onler and Burak Ozdonglar for the roughness image of the 3D printed cylinder. This work was partially funded by America Makes and Rice University.

References

Asadi-Eydivand, M. et al., 2016. Optimal design of a 3D-printed scaffold using intelligent evolutionary algorithms. *Applied Soft Computing Journal*, 39, pp.36–47.

Basheer, I.A. and Hajmeer, M., 2000. Artificial neural networks: fundamentals, computing, design, and application. *Journal of microbiological methods*, *43*(1), pp.3-31.

Beuth, J., Fox, J., Gockel, J., Montgomery, C., Yang, R., Qiao, H., Soylemez, E., Reeseewatt, P., Anvari, A., Narra, S. and Klingbeil, N., 2013. Process mapping for qualification across multiple direct metal additive manufacturing processes. In *Proceedings of SFF Symposium.,*

Austin, TX, Aug (pp. 12-14).

Bharadwaj, R., 2012. Using DEM to solve bulk material handling problems. *Chemical Engineering Progress*, 108(9), pp.54–58.

Bishop, C.M., 2006. Pattern recognition. *Machine Learning*, *128*, pp.1-58.

Chen, H. and Zhao, Y.F., 2015, August. Learning Algorithm Based Modeling and Process Parameters Recommendation System for Binder Jetting Additive Manufacturing Process. In *ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (pp. V01AT02A029-V01AT02A029). American Society of Mechanical Engineers.

Dougherty, P., 2016. Synthesizing in situ Friction and Wear with ex situ Surface Metrology to Provide Post-mortem Tribological Analysis: Experiments and Modeling.

Ferrez, J.A., 2001. *Dynamic triangulations for efficient 3D simulation of granular materials* (Doctoral dissertation, ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE).

Gockel, J., Beuth, J. & Taminger, K., 2014. Integrated control of solidification microstructure and melt pool dimensions in electron beam wire feed additive manufacturing of Ti-6Al-4V. *Additive Manufacturing*, 1–4(4), pp.119–126.

Green, S., 2010. Particle simulation using cuda. *NVIDIA whitepaper*, *6*, pp.121-128.

Haeri, S., Wang, Y., Ghita, O. and Sun, J., 2017. Discrete element simulation and experimental study of powder spreading process in additive manufacturing. *Powder Technology*, *306*, pp.45-54.

Herbold, E.B., Walton, O. and Homel, M.A., 2015. *Simulation of Powder Layer Deposition in Additive Manufacturing Processes Using the Discrete Element Method* (No. LLNL--TR-678550). Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States).

Ho, C.Y. and Lee, Y.C., 2007. Temperature fields in the fusion zone induced by a moving electron beam. *Journal of mechanical science and technology*, *21*(10), pp.1707-1713.

Huang, S.H., Liu, P., Mokasdar, A. and Hou, L., 2013. Additive manufacturing and its societal impact: a literature review. *The International Journal of Advanced Manufacturing Technology*, pp.1-13.

Hu, L., Hu, G.M., Fang, Z.Q. and Zhang, Y., 2013. A new algorithm for contact detection between spherical particle and triangulated mesh boundary in discrete element method simulations. *International Journal for Numerical Methods in Engineering*, *94*(8), pp.787-804.

Jiang, Z., Liu, Y., Chen, H. and Hu, Q., 2014, September. Optimization of Process Parameters for Biological 3D Printing Forming Based on BP Neural Network and Genetic Algorithm. In *ISPE CE* (pp. 351-358).

Mishra, B. K., and C. V. R. Murty. "On the determination of contact parameters for realistic DEM simulations of ball mills." *Powder Technology* 115.3 (2001): 290-297.

Mindt, H.W., Megahed, M., Lavery, N.P., Holmes, M.A. and Brown, S.G.R., 2016. Powder bed layer characteristics: the overseen first-order process input. *Metallurgical and Materials*

Transactions A, *47*(8), pp.3811-3822.

Mpagazehe, J.N., 2013. A Physics-Based, Eulerian-Lagrangian Computational Modeling Framework to Predict Particle Flow and Tribological Phenomena.

NVIDIA, C. U. D. A. "Programming guide." (2008).

Parteli, E.J. and Pöschel, T., 2016. Particle-based simulation of powder application in additive manufacturing. *Powder Technology*, *288*, pp.96-102.

Passerello, C.E., 1982. Interference detection using barycentric coordinates. *Mechanics Research Communications*, *9*(6), pp.373-378.

Tourloukis, G. et al., 2016. Predictive Modelling for 3D Inkjet Printing Processes. *17th International Conference on Thermal, Mechanical and Multi-Physics Simulation and Experiments in Microelectronics and Microsystems (EuroSimE)*, pp.257–262.