Jack P. Lombardi III
Department of Systems Science and Industrial Engineering,
Binghamton University,
Binghamton, NY 13902
e-mail: jlombar4@binghamton.edu

Roozbeh (Ross) Salary
Division of Mechanical Engineering,
Marshall University,
Huntington, WV 25755
e-mail: salary@marshall.edu

Darshana L. Weerawarne
Center for Advanced Microelectronics Manufacturing,
Binghamton University,
Binghamton, NY 13902
e-mail: dweerawa@binghamton.edu

Prahalada K. Rao
Department of Mechanical and Materials Engineering,
University of Nebraska-Lincoln,
Lincoln, NE 68588
e-mail: rao@unl.edu

Mark D. Poliks[1]
Department of Systems Science and Industrial Engineering,
Binghamton University,
Binghamton, NY 13902
e-mail: mpoliks@binghamton.edu

Image-Based Closed-Loop Control of Aerosol Jet Printing Using Classical Control Methods

Aerosol jet printing (AJP) is a complex process for additive electronics that is often unstable. To overcome this instability, observation while printing and control of the printing process using image-based monitoring is demonstrated. This monitoring is validated against images taken after the print and shown highly correlated and useful for the determination of printed linewidth. These images and the observed linewidth are used as input for closed-loop control of the printing process, with print speed changed in response to changes in the observed linewidth. Regression is used to relate these quantities and forms the basis of proportional and proportional integral control. Electrical test structures were printed with controlled and uncontrolled printing, and it was found that the control influenced their linewidth and electrical properties, giving improved uniformity in both size and electrical performance. [DOI: 10.1115/1.4043659]

1 Introduction

1.1 Objective and Motivation. Aerosol jet printing (AJP) is a high-resolution printing technique for additive fabrication of electronics. Among the advantages of this technique are high resolution, a wide viscosity range of materials, and a large standoff distance from the substrate. This technique has been used for the creation of various electrical devices, from interconnects and components [1–6] to sensors [7,8], and even printing electronics directly on objects [9]. Because of these advantages, work is being done to improve the AJP process and allow for high quality, more consistent prints. Accordingly, the objective of this work is to develop and implement an in situ image-based monitoring and closed-loop control approach to ensure print quality in AJP. This work thus takes the first step toward the long-term goal that changes in the print output may be sensed and compensated for while printing.

1.2 Background and Gaps. Currently, studies of AJP have been done to characterize the printing process and optimize the printed output, specifically AJP lines that are used to make traces and structures. Mahajan et al. [10] described how printing with the ultrasonic atomizer could be optimized and how the ratio of sheath and carrier flows affects the formation of thick, narrow printed lines. Verheecke et al. [11] demonstrated the optimization of AJP lines using the pneumatic atomizer and rudimentary image processing to optimize for narrow printed lines. Likewise, Goth et al. [12] also optimized AJP using the pneumatic atomizer, conducting experiments at different flows and observing the resulting linewidths to determine process parameters. Salary et al. [13,14] also aimed to optimize AJP using visual quantifiers to indicate line quality, which was confirmed through electrical measurements verifying print quality. Thompson and Yoon [15] discussed how AJP quality can be optimized through toolpath planning and ensuring a constant velocity during the printing process, as the print speed is critical, but do not incorporate any control or monitoring into their process. Smith et al. [16] also described the optimization on different substrates using AJP with different process parameters and also included an assessment of process drift. Gu et al. [17] provided an assessment of AJP printing capability using microwells with a known volume, which are filled using printing to show the rate of AJP material deposition and allow for the characterization of the printing process. Sun et al. [18] and Li et al. [19] also investigated fitting complex statistical models to AJP quality based on image data.

These characterization and optimization studies do not take into consideration that AJP is not a static process and that it tends to drift over time. During AJP, the observed output may drift or fluctuate, leading to irregularities in the output even though the print parameters have not changed. Additionally, they are all based on off-line (postprinting) characterization data, assuming that the optimal process operability window obtained from an experiment can be usefully applied to subsequent experiments. Due to these shortcomings in detection and compensation in print fluctuations, we developed a new vision-based method for the monitoring, optimization, and control of AJP. This method makes use of the standard process monitor camera on the Optomec AJ-300 aerosol jet printer to observe print quality during the print and image processing to provide information about the print quality while printing. From this data stream, changes can then be made to the print

[1]Corresponding author.

Manuscript received July 23, 2018; final manuscript received April 10, 2019; published online May 28, 2019. Assoc. Editor: Laine Mears.

parameters, such as process flows or print speed, and a more consistent print can be obtained. Specifically, we alter the print speed to compensate for changes in ink output and ensure a consistent deposition rate of material based on print speed. Though this sort of closed-loop control for deposited material is new for AJP, it has been previously used for additive manufacturing based on welding processes. In the work by Xiong and Zhang [20], a controller using image data and extracting the distance between a welding gun and workpiece was demonstrated. It was found that this control could compensate for deposition and height variations and give an output closer to a specified value. Xiong et al. [21] also characterized the response of the weld process using a system identification process and subsequently designed a vision-based controller to vary the weld travel speed to control for changes in welding parameters such as current and temperature. This control was also used to vary the width of the weld, and a part was demonstrated with variable thickness, created by varying the weld speed.

These works show an analogous process that was able to be controlled and optimized using visual data. We will follow a similar approach for our control of AJP linewidth, varying print speed to control linewidth. Likewise, tests are done with multiple speeds to create data on the relationship between linewidth and print speed. Regression is used to create a relationship between print speed and linewidth, and we assume that the width is a direct function of print speed at a given material output. If the material output changes due to drift or some other change in the printer, and the linewidth increases, this change can be compensated for using the same relationship we observe in changing the print speed. This relationship is used to create a simple proportional controller. The regression slope is used as a basis for the controller gain, with the reciprocal of it being used as the gain. Control inputs come from the error between a specified linewidth, which is observed for the initial print parameters, and the currently observed linewidth. The controller also has a tolerance or deadband, as well as a function to ignore unusually large changes in linewidth, as these are anomalies. If an error is larger than the tolerance, it is divided by the regression slope to find a new print speed to increase or decrease the linewidth back to the specified width. Given the slow, linear nature of the drift, as described by Smith et al. [16], proportional control with the tolerance defining a deadband could be adequate to improve the AJP process and demonstrate closed-loop control [22]. This monitoring and control is implemented through software, and a few different languages and software packages are needed to handle the communication, image processing, and execution of the print process. This includes PYTHON-OPENCV, C++, ACSPL+ (language of the ACS motion controller), and KEWA (Optomec AJP control software). With the implementation of the forwarded vision-based sensing and control approach, we (i) validate the ability to sense linewidth in near real-time compared with in-line measurements; (ii) create a regression model of linewidth versus print speed; and (iii) demonstrate closed-loop control of printed linewidth using proportional (P) as well as proportional-integral-derivative (PID) controllers. Test structures are printed and compared with those printed without control to assess the controller's impact on AJP quality. After initial experiments and results using this controller, shortcomings were identified, and it was decided that the design of a PID controller could further improve the performance [22]. Design of a new controller used a simulation-based approach with MATLAB SIMULINK used to simulate the printer behavior with a PID controller. Using the SIMULINK model, a proportional integral (PI) controller was designed and implemented. Test structures were again printed and compared with those printed without control, and the PI controller was found to improve AJP quality, as compared with uncontrolled prints and the previous controller. The following sections describe the experimental setup, with the materials used and configuration of the printer for control, followed by the results of the experiments and measurements of the printed structures. The final section summarizes and concludes this work.

2 Experimental Setup and Materials

2.1 Substrate and Ink. All printing was done on Ube Plastics UPILEX 125S polyimide, which was used as received. Paru PG-007AP silver nanoparticle ink was used in the pneumatic atomizer of the Optomec AJ-300 printer. This ink contains silver nanoparticles with an average size of 100 nm.

2.2 Camera Configuration. The AJ-300 printer includes two cameras, one for process monitoring, observing the spot of deposition, and one for alignment, allowing visual location and inspection of features on the substrate and setting of the print origin. Both are used in this study and are standard for the Optomec AJ-300 system, with the process monitor camera for in situ images and the alignment camera for validation images. Both cameras are configured and controlled using a custom Windows application written in C++ using Point Grey's FLYCAPTURE2 application programming interface (API). Both cameras are set to free running mode with frame rates of 50 fps. The start/stop of the image capturing is triggered using memory buffer flags of the motion controller set/reset in synchronization with the printing start/stop.

The process monitor camera, which is inclined to look at the point of ink deposition at approximately 45 deg, was used to capture images of the printed line while it is being deposited. All prints are made in the *y*-direction of the stage, so that the line is always in-line with the camera, and the imaged area is in focus. The process monitor camera consists of a Point Grey Flea3 1.3 MP Color GigE camera, with an Infinity Photo-Optical InfiniStix™ 1.0X, 94 mm working distance video lens and 4X magnification module. The camera was restricted to 300×100 pixels to observe an approximately $300 \times 100\ \mu$m section of the printed line with 1.01 μm/pixel resolution, with an exposure time of 1 ms and frame rate of 50 fps. Narrowing the camera window size (also known as windowing) serves two functions: (i) it restricts the image to only the depth of field of the camera and (ii) it reduces the image size, which allows for faster image processing. Images are saved as raw12 data format and .tif file format. A Dolan-Jenner PL-800 fiber coupled lamp is used to illuminate the printed line. It is mounted opposite of the camera to provide the greatest amount of reflected light to it. A paper diffuser is mounted on the end of the fiber, and the fiber is positioned so that the background of the image is saturated, aiding in the identification of the linewidth. The layout of this camera and illumination is shown in Fig. 1, and an example image in Fig. 2(*a*).

The alignment camera is used to look at the printed line after it is printed and provide validation information on the width of the printed line. This camera is mounted perpendicular to the platen and consists of a Point Grey Flea3 1.3 MP Color GigE camera mounted to an Infinity Photo-Optical InfiniTube™ FM-200, which has coaxial illumination and a 4X, 18 mm working distance objective, which provides a total magnification of 8X. The camera was windowed to 720×480 pixels to observe a $150 \times 100\ \mu$m section of the printed line with 0.48 μm/pixel resolution, with an exposure time of 1 ms and frame rate of 50 fps. Images are saved as raw12 data format and .tif file format. Illumination is provided by a Philmore 80 Lumen, 6000 K color LED placed on the coaxial illumination port. The layout of this camera can be seen in Fig. 1 and an example image in Fig. 2(*b*).

2.3 Image Processing and Analysis. A major component of this effort is the need to analyze the collected images and extract morphology data from them to be used for analysis and control while printing. In our previous work [13,14], we successfully implemented algorithms for in-line analysis. These intensity-based linewidth algorithms were optimized and implemented in PYTHON-OPENCV, a compilation of image processing functions and algorithms executable from PYTHON. The PYTHON-OPENCV implementation provides us the capability of high-speed image processing in our application. The computational complexity and the workload of the image processing

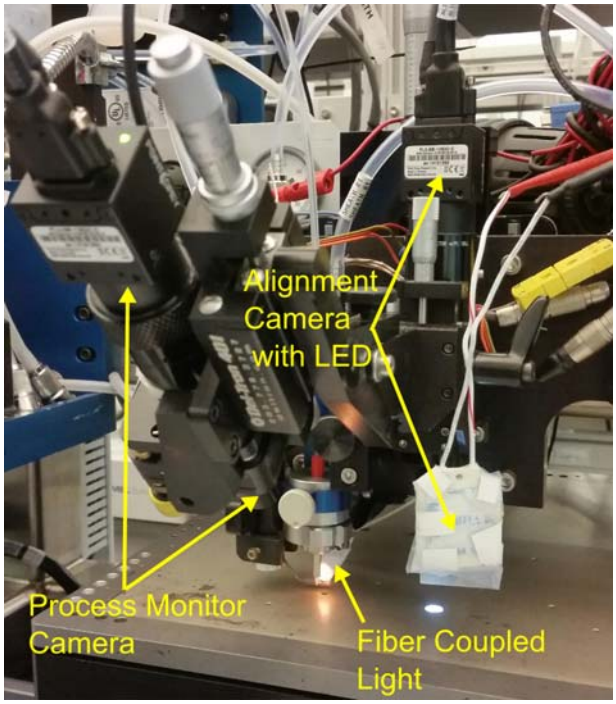

Fig. 1 Photo of process monitor and alignment cameras on the Optomec AJ-300 used in this study

were optimized such that enough computational resources were allocated for other control project-related applications.

The algorithm for finding the linewidth starts by importing a saved image (I), $I \in \mathbb{R}^{m \times n}$, in gray-scale format into PYTHON-OPENCV, where it is initially complemented, transposed, and normalized (to be globally mapped in the range of [0,1]). We note that the background is saturated during printing in all experiments. The imported images are then dynamically cropped to prevent oversampling. Subsequently, in order to remove noise and obtain a smooth profile of intensity, the pixel values of each row are averaged and stored as a new element in a vector (J), $J \in \mathbb{R}^{n \times 1}$. The resulting vector reveals the regions of line and background more smoothly. Finally, the algorithm searches for vector indices whose row-wise averaged intensity values are less than or equal to a set threshold. After the locations of the edges are found, the difference between them is calculated for the width. The quantified line width is logged and shared with other modules of the code to execute control. This method has appeared to be accurate, efficient, and quick for process control. It is worth mentioning that we have developed several other image processing algorithms (although not used in this work) that can be utilized for edge detection in various process and illumination conditions. An example of edge detection on an image is shown in Fig. 3.

As mentioned above, image processing is used to solve the problem of oversampling the printed line. Oversampling results from the fact that an observed line in an image is 100 μm long, and at low print speeds, such as 2 mm/s, the printer will only travel 40 μm between images, meaning 60% of the image was already sampled for linewidth determination, as shown in Fig. 3. To avoid this, a routine is implemented to crop images based on print speed, removing the previously sampled areas from the analysis. This allows the use of a constant sampling rate while preventing oversampling of the printed line with the changing speed. It is known though, that operating at this frame rate will result in undersampling at higher speeds, as the printer will travel 200 μm between images at 10 mm/s. This is of course undesirable, but unavoidable, as the frame rate cannot be increased further at this time. The results of this cropping are shown in Fig. 3.

3 Communication Protocols, Synchronization, and Control

3.1 Communication Protocol. A custom Windows program written in C++ communicated to the cameras and the motion controller. The camera communication is implemented using Point Grey's FLYCAPTURE2 C++ API. The functionalities of the application include registering camera settings, image capturing, and file saving. The communication to the ACS motion controller is implemented using ACS SPIIPLUS C++ API. The application is capable of

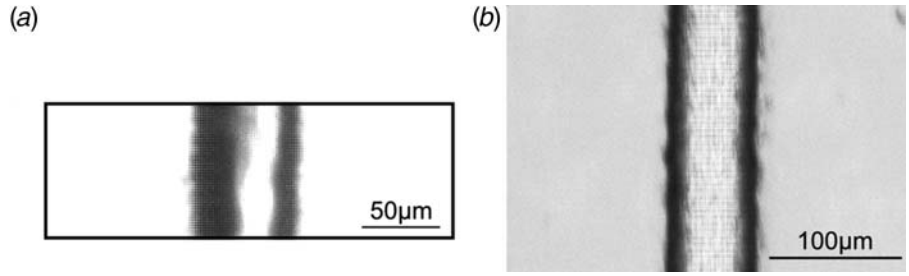

Fig. 2 Example images collected in this study from the process monitor (*a*) and the alignment camera (*b*). Note that the shading in the process monitor image is due to the lighting and observation at an angle.

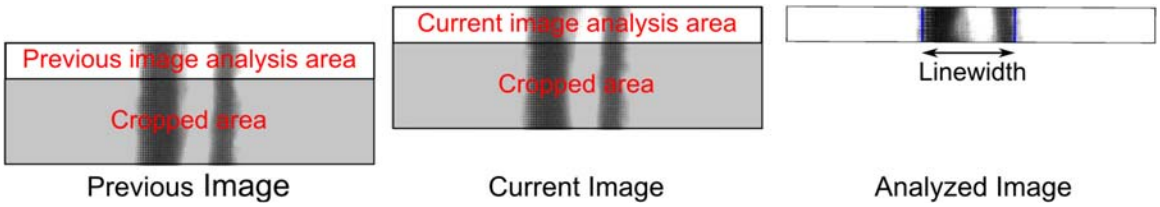

Fig. 3 Graphic depiction of cropping and image analysis at 2 mm/s print speed. The previous image is shown on the left, and the area analyzed and cropped from that image is removed from the current image. The current image is then cropped an analyzed, with the detected line edges shown in blue.

detecting the axis motion, reading/writing the axis speed, and reading the global buffer of the motion controller.

3.2 Synchronization. The system requires the C++ Windows application and the PYTHON script (used for linewidth detection and print speed calculations) to run simultaneously. The interprocess communication/synchronization is accomplished via shared files accessible by both programs. The C++ application writes to a *flag file* setting/resetting the motion flag depending on when the axis motion starts/stops. The PYTHON-OPENCV application reads this file and starts/stops image processing based on it. It then writes a calculated speed to a *speed file* specifying the target speed of the axis. The C++ application reads this file and sets the axis motion speed accordingly. Both C++ and PYTHON applications constantly and appropriately check these file contents and change the program behavior accordingly. Operating system level file access control allows proper read/write synchronization. Furthermore, proper error handling has been incorporated into the applications to avoid possible file access errors. The Windows C++ application uses POSIX Threads (Pthreads) to run image capturing and motion controlling in separate threads to avoid any delays introduced by a sequential execution of two functions.

3.3 Control. Both C++ and PYTHON-OPENCV are running before a print in the *waiting* mode. Next, we start the printing by running an ACSPL+ file through KEWA. The ACSPL+ file contains the toolpath for the print, and it sets/resets a user-defined global flag variable when it wants the camera to start/stop capturing the images. As soon as the motion starts, the C++ application sets the motion flag file, driving the PYTHON application to *ready* mode and starts reading the user-defined global flag variable in the buffer of the motion controller. It then starts/stops image capturing in synchronization with the global flag variable value. As soon as the images are available, the PYTHON application analyses the images, applying the cropping and processing methods discussed earlier. PYTHON-OPENCV may then make changes to the speed file based on the images read in. Any changes in the speed file are read by the C++ application for updating the motion speed.

The previous general description is how all the pieces work together to collect data during a print. This process forms the basis for the data collection in calibrating the process, finding the controller gain, and implementing the control. For collecting information on the gain, the relationship between the linewidth and print speed, the print speed is varied based on the number of images captured, giving the ability to collect data on the linewidth for different speeds in one experiment. After this experiment, a PYTHON code is used to quickly analyze the data and find the gain by selecting 30 points from each speed and performing linear regression on the selected data. This gain value is also written to a text file to allow it to be accessed for controlled printing.

The speed calculation is also implemented in PYTHON-OPENCV and is similar in its implementation of writing a calculated new print speed to a file that will be continuously read by the C++ application and fed into the motion controller. The difference is that the speed is being computed with each photo that is taken based on the linewidth value. This linewidth value is logged and also passed to the algorithm for control. The algorithm for the speed control also consists of a few logical operations to create a tolerance or deadband, remove anomalous linewidths, and prevent extreme print speeds. The algorithm consists of first estimating if the difference between the specified linewidth and observed linewidth is greater than the deadband and less than the anomalous result threshold. If either of these conditions is violated, then the linewidth is rejected and the print speed is unchanged. If the linewidth is within the window of error, a new print speed is calculated based on Eq. (1). That is, by subtracting the original specified linewidth from the observed linewidth, an error signal is produced. This is then multiplied by the reciprocal of regression slope (the proportional gain), to give a correction speed to be added to the original print speed, creating the new print speed.

$$\text{New print speed} = \text{default print speed} + \frac{\text{error}}{\text{regression slope}} \tag{1}$$

where error is the difference between the specified and observed linewidth. This new speed is then compared with maximum and minimum print speeds, and if it is between them, it is written to a file for the C++ application to implement, otherwise the previous speed is maintained and if no history is recorded, it is set to the nominal value of 4 mm/s.

The proportional integral control is implemented in a similar way and uses PYTHON-OPENCV to calculate the correction added to the default print speed based on current and integrated error, as shown in the standard form of the proportional integral controller in Eq. (2)

$$\text{New print speed} = \text{default print speed} + K_p\left(\text{error} + \frac{1}{T_i}\int \text{error}\right) \tag{2}$$

where K_p is the proportional gain and $1/T_i$ serves as the integral gain [22]. The integral of the error is calculated using the trapezoidal rule and logged in the PYTHON-OPENCV program. The same protections for anomalous linewidths and minimum and maximum speed are also kept.

3.4 Automated Test Printing. Given the complicated, multicomponent nature of this experiment, it was decided that the best way to implement the printing of test structures was to automate the process. This allows for the printing of test structures quickly and uniformly, with a random pre-allocation of whether or not they are printed with control, based on a text file read by C++ and PYTHON-OPENCV, with data being collected throughout the printing process. The ACSPL+ file was modified to print multiple four-point or Kelvin structures, which is done by adding a loop to add an offset to the toolpath with each iteration. While the printer is executing the ACSPL+, C++, and PYTHON-OPENCV are running as well, with the C++ code saving images based on the flag variable in the ACSPL+ file for image collection, and changing the print speed based on text file value written by PYTHON-OPENCV. The C++ will also create a new folder to save the images for each new print based on the file specifying prints, and PYTHON-OPENCV will read and analyze those images, changing the speed for control, and saving linewidth data to a text file once motion stops. The relationship among the control applications (software programs) is summarized in Fig. 4, which shows all the components/blocks of the proposed closed-loop control platform as well as the flow of data among them.

4 Experimental Procedure

The experimental execution of this work can be split into two parts: the validation of the in situ monitoring and the implementation and execution of printing with closed-loop control. The procedure used to validate the monitoring will also serve to find the gain used for the proportional controller in the closed-loop control experiment. All printing experiments used the pneumatic atomizer with the Paru ink, the 200 μm tip and 3 mm standoff from the polyimide substrate. The print parameters of 40, 600, and 580 sccm were used for sheath, atomizer, and exhaust flows, respectively, and resulted in a line of approximately 70 μm wide at 4 mm/s print speed, the baseline for these experiments. The experiment consisted of printing the test for linewidth as a function of speed, after a 5-min equilibration period, followed 2-min later with the first set of four-point structures. Each set consisted of ten structures. Two more sets of four-point structures were printed, 10-min apart, 5 min printing and 5 min waiting. This brought the total print time to be approximately 40 min, a generally accepted runtime for the printer.

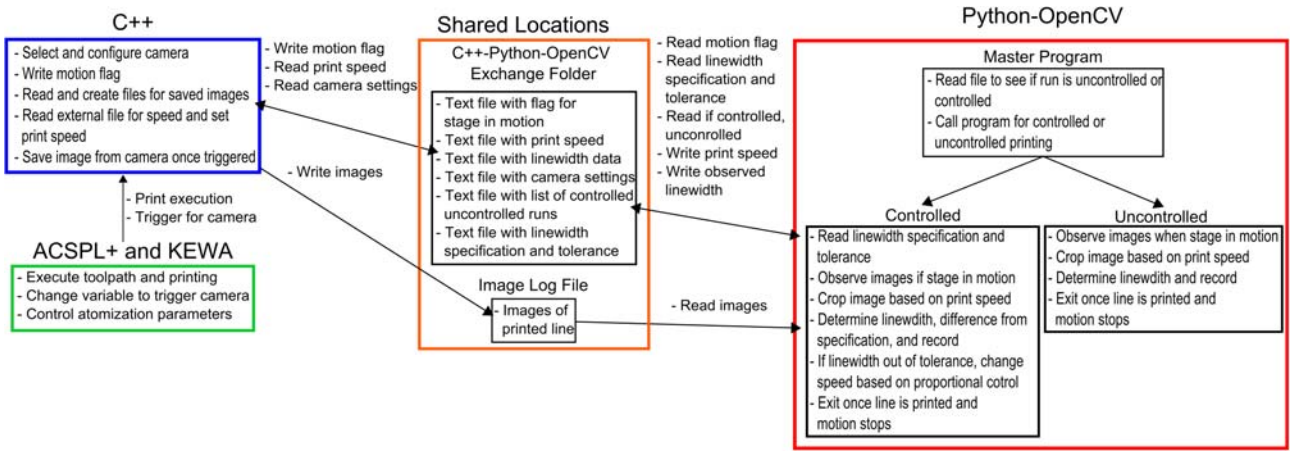

Fig. 4 Diagram of the various inputs, outputs, and software used to monitor and control the AJP process

4.1 In Situ Monitoring Validation. In this experiment, we prove that the results given by the process monitoring camera while printing are nearly identical to results gathered from the analysis of images taken after the fact. The procedure was to create an ACSPL+ file to print a line 50 mm long, and program the PYTHON-OPENCV to change the speed file (therefore the print speed) after 50 images were taken at randomly assigned print speeds of 4, 2, 10, 6, and 8 mm/s. After this line is printed, it is then observed using the alignment camera, running the same ACSPL+ file, with speed file modified to run with the speed at 4 mm/s the whole length, and the parameters for image analysis changed to analyze these different images. The linewidths recorded from these two observations were then compared to validate the process monitor camera linewidths.

The linewidth data from the process monitor camera was then used to create the regression model between the linewidth and the print speed. From this linewidth data, 30 points were taken from each observed print speed and used for the fitting of the regression model. These same points were used for both the linear regression used for the P controller and later the polynomial regression used for the creation of the SIMULINK model in design of the PI controller.

4.2 Printing of Four-Point Test Structures. Once the controller gain was determined from the in situ validation test, printing of four-point test structures could be done. These structures were used to accurately test electrical resistance, and a layout of one is shown in Fig. 5. Their fabrication was done in two steps: first, the printing of the conducting line, either at the default speed or with control, then the printing of the pads for all the structures.

The run order was randomized, with ten structures printed, five controlled and five uncontrolled. This run of ten structures was done three times, with the start of each set 10-min apart. This is done to assess the ability of the control to compensate for drift over time. After printing, all samples were thermally sintered by baking in a Binder convection oven at 200 °C for 60 min, with a 20-min ramp to temperature. Electrical measurements were made with a Suss probe station connected to a Keithley 2614 source measure unit. A current of 10 mA was used for the four-wire resistance measurement of the printed structures. This procedure for printing and testing was conducted for both the proportional and proportional integral controllers.

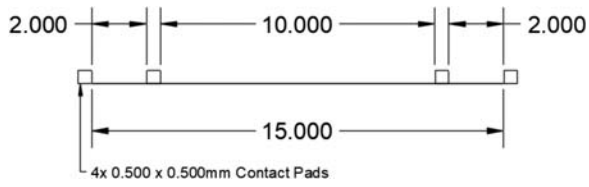

Fig. 5 Layout of the four-point test structures. The central line in the structure was printed controlled or uncontrolled.

5 Results and Analysis

First, the observations of the process monitor camera are validated, and the regression model based on these observations and P controller is created. Experiments using the P controller are described, followed by the creation of the SIMULINK model and PI controller, and lastly, results of experiments with that controller.

5.1 Validation of the In Situ Monitoring Data. The first experiment performed after the printer was started was to run a print with different speeds, making different linewidths. The purpose of this is twofold: to create the relationship between print speed and linewidth and to provide validation for the new in situ observation method.

A plot of the linewidth data for each observation method is shown in Fig. 6. In Fig. 6, the average linewidth and standard deviation for each print speed are shown, based on a sample of 30 points at each print speed. A strong correlation of $\rho = 0.993$ is seen between the two plots, indicating that they are seeing the same

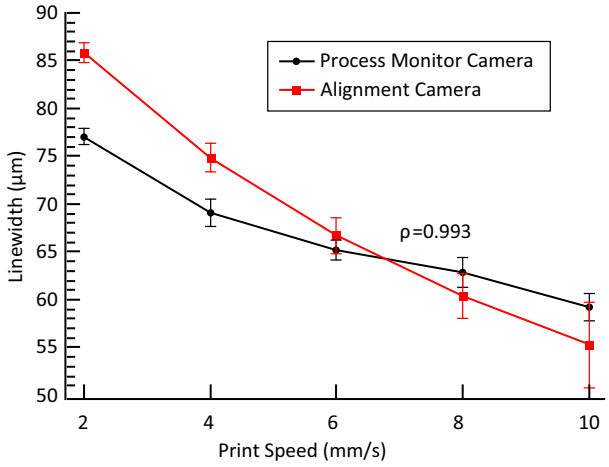

Fig. 6 Comparison of observed linewidths at different print speeds by the process monitor and alignment camera. The results are highly correlated and similar. The error bars represent one standard deviation.

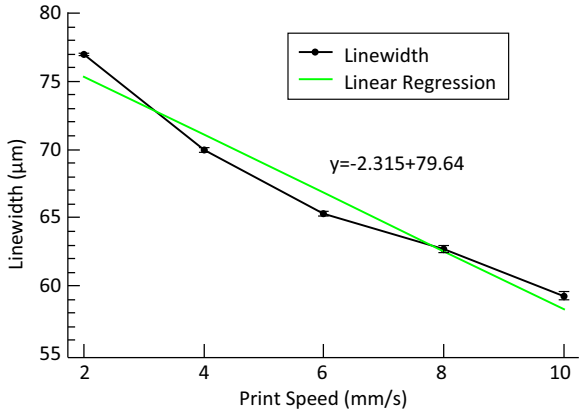

Fig. 7 Plot of observed linewidth as a function of print speed, used to find proportional controller gain. The error bars show the standard error of the mean.

changes in linewidth due to print speed. While the observations are strongly correlated, there are still differences in measurements. This is likely due to the less defined edges in the process monitoring camera images, as shown in Fig. 2, and also slight changes in the image intensity due to print speed. The reflectance and texture of the ink can also change due to drying, and this change could be perceived as different widths. Evidence for this is the fact that the large, very wet 2 mm/s prints and the thin, very dry 10 mm/s had the largest discrepancies, while the intermediate prints had less, as shown in Fig. 6.

A final analysis of this data is performed to determine the gain to be used in the proportional controller, performing a regression analysis with 30 linewidth points from each print speed. The plot shown in Fig. 7 shows this data and the regression, which had an R^2 value of 0.962 and slope of −2.315. The reciprocal of this slope was then used as the gain for the proportional controller. These experiments validate the use of the process monitoring camera for monitoring of the linewidth. We approximate a linear relationship between the linewidth and print speed for simplicity in this attempt. This serves as a basis for implementing proportional control.

5.2 Comparison of Controlled and Uncontrolled Four-Point Structures. Before printing with and without control, the proportional controller had to be implemented and updated. This consisted of setting the anomaly rejection threshold, tolerance for linewidth, and proportional gain. The terms of anomaly rejection threshold and tolerance serve to prevent changes in print speed due to erroneously large linewidths from contamination, such as dust particles, or excessive control effort being applied to nondrift variation in the linewidth. The anomaly rejection is set to 15 μm, tolerance to 1.5 μm, and gain to −2.315. The anomaly rejection and tolerance were set based on previous experience, while the gain was set based on the previous print speed experiments. The first set of printed four-point structures is shown in Fig. 8. The average linewidths for the five controlled and uncontrolled samples in each set are shown in Fig. 9. While the first set shows little difference between controlled and uncontrolled, later sets, when changes in the printer output occurred (i.e., when drift has become significant), show a definite difference between the controlled and uncontrolled samples in each set. This shows that the control is having an effect and changing the linewidth in response to observation. When these data are compared with the original specification made at the beginning of the print for a linewidth of 70 μm, it is also clear that the controlled samples remain closer to that specification. Although the controlled linewidths show larger standard deviations within each set, they show less variation among the three sets as compared with the uncontrolled sets, as can be seen by comparing the averages of the three controlled sets. This is likely due to oscillations in the

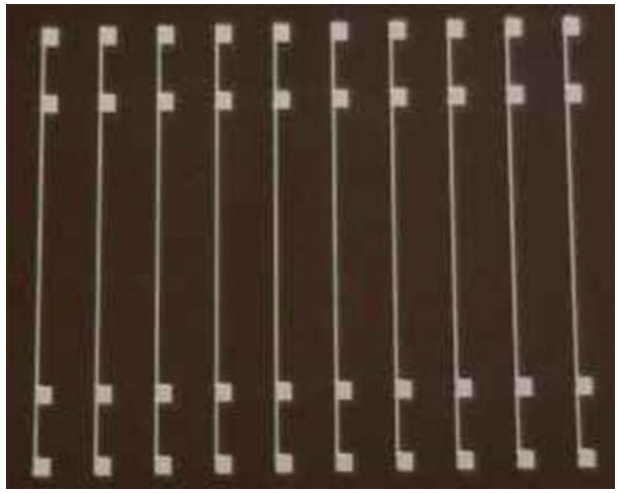

Fig. 8 Photograph of the first set of four-point structures. The dimensions of the four-point test structures are shown in Fig. 5.

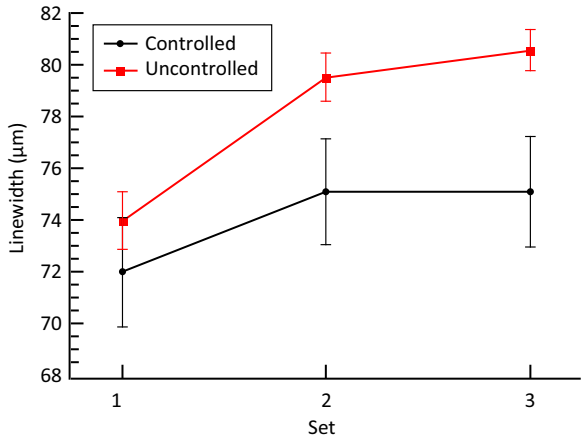

Fig. 9 Plot of average linewidth for the controlled and uncontrolled four-point structures using the P controller. The error bars represent one standard deviation.

control system, an example of which is shown in Fig. 10, where the linewidths in the third controlled and uncontrolled structures in the second set of prints are plotted, where the linewidth will vary in a regular pattern only in the controlled case.

This oscillatory behavior is typical for a proportional controller and demonstrates that closed-loop control is initiated [22]. Additionally, it can be noted that the print speed has not reached its maximum even though the line is continually out of specification. This is likely due to the print output continuing to increase, such that the initial regression curve is no longer valid, and the print speeds determined by that initial curve are unable to bring the linewidth back into specification. Due to these issues with the P controller, a PI controller was developed, as described in Sec. 5.3.

Electrical resistance data are shown in Fig. 11, from which a clear difference can be seen between the controlled and uncontrolled samples. While the controlled samples all had a higher resistance, due to faster print speeds and less deposited material, they were also more consistent, with an overall average of 7.158 Ω and standard deviation of 0.3579 Ω over the three sets. The uncontrolled, however, has an average of 5.019 Ω and standard deviation of 0.7029 Ω across the three sets. The lower observed resistance is due to the uncontrolled print settings creating oversized lines that then have a lower resistance. This shows that the control method was able to affect the electrical properties of the observed resistance.

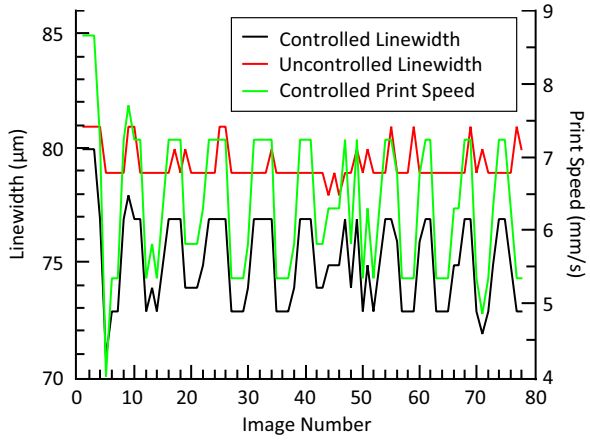

Fig. 10 Plot of linewidth and print speed for the third controlled and uncontrolled four-point structures printed in Set 2. Note how the controlled data has a regular oscillation and is closer to the specified linewidth.

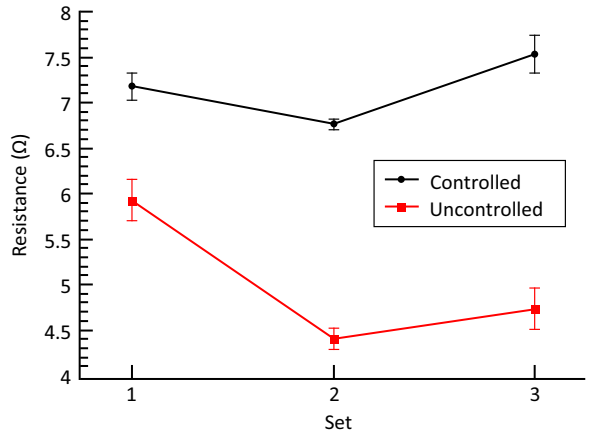

Fig. 11 Plot of average electrical resistance of the controlled and uncontrolled four-point lines using the P controller. The error bars represent one standard deviation.

Overall, comparing the results of controlled and uncontrolled samples, we see that although the uncontrolled sets have smaller in set variation, as shown by the smaller error bars in Fig. 9, the average linewidth changes from 74 μm to 81 μm in the uncontrolled case as compared with 72–75 μm in the controlled case over the course of printing. A similar result is seen in Fig. 11, where the controlled sets have a higher, but more consistent resistance. While these results show that there was an improvement, the specification for linewidth was not met in the second and third sets, and the controller did not increase print speed to its maximum to meet this. As discussed, this is likely due to a changing linewidth-print speed relationship, with the slope and gain computed at the beginning of the print no longer being relevant. This is backed up by the fact that the observed print speeds, such as the example in Fig. 10, are accurate to that initial setting; however, the change in the material output causes these speeds to not have the same effect in decreasing linewidth. Further study of this change in the print speed-linewidth relationship and development of more sophisticated control algorithms, such as PID control, could minimize this problem and are described below.

5.3 Improvements in Modeling and Control. Given the shortcomings of the previous P controller, other controllers would need to be investigated, such as the common proportional integral (PI) controller. However, the challenge was finding the appropriate value of the gains to be used in this controller. This in turn would require a plant model of the printer for the linewidth and print speed; however, this is not as trivial as anticipated due to the nonlinearity of the system response. This nonlinearity is due to the fact that the motion controller sampling frequency and speed changes occur very rapidly, on the order of milliseconds, while the camera can only acquire images every 20 ms. This means that in a conventional system identification *bump test*, where a step change is made to the input and the output is observed, there are no observed intermediate linewidths, only the steady-state linewidths at each speed, with a 100 ms delay between the two values. This 100 ms delay is an overall delay due to image acquisition, storage, and processing time, as well as offset in observation from the point of print to where the line is imaged. This implies that conventional transfer function models and system identification could not be used to create an accurate plant model. Since a conventional model could not be used, we instead used the data on linewidth as a function of print speed that was shown in Fig. 7 and used polynomial regression to create a model of the printer output. This equation is shown in Eq. (3)

$$\text{Linewidth} = -0.055\text{PS}^3 + 1.171\text{PS}^2 - 9.478\text{PS} + 91.777 \quad (3)$$

where PS is print speed and was found to accurately model the average linewidth at each print speed and attempt to accurately represent the behavior of the printer. Using this equation and the observed delay, a model for the plant and control could then be made in MATLAB using SIMULINK. A screen capture of this model is shown in Fig. 12.

It was found that this model could accurately replicate the linewidth of the printer and demonstrate closed-loop behavior, such as oscillation. Using this model for the printer and controller, a proportional-integral-derivative (PID) controller was tuned using the MATLAB PID Tuner App, and the derivative term was found to be 0, making the controller a PI controller. After initial gains were found, the robustness of the control was tested by adding in ramp inputs to simulate the drift of the printer and Gaussian noise to simulate the normal variation in linewidth. This model was also tested with different linewidth specifications, such as 65 and

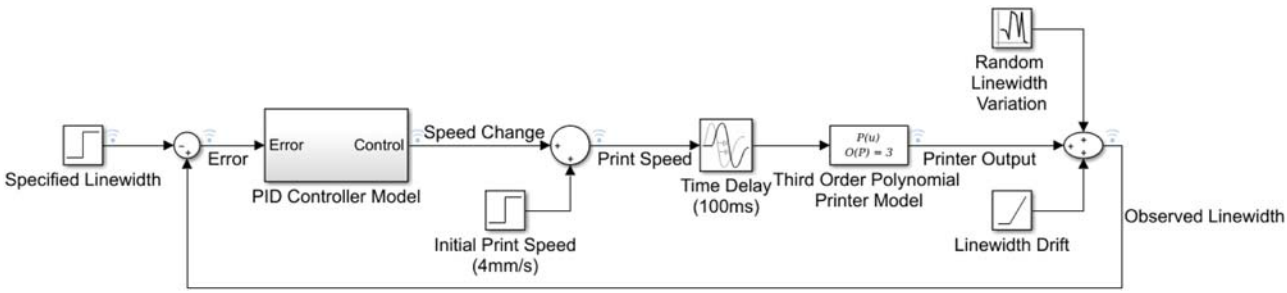

Fig. 12 A screen capture of the MATLAB SIMULINK model used to test PID controller designs and create a PI controller that was then implemented on the printer

80 μm, and was found to control the print speed to allow for those linewidths.

This PI controller was then implemented on the printer, with the P controller code modified to now numerically integrate the observed error and use that to help control the linewidth based on the print speed. Experimental tests of this controller showed that it worked, and manual tuning to improve performance gave proportional gain K_p of 1.055e^{-5} and integral time T_i of 6.667e^{-6} when written in the standard form. The same experiment using printed four-point structures was then carried out to assess the impact of the new PI controller on the linewidth and resistance of the printed lines.

5.4 Comparison of PI Controlled and Uncontrolled Four-Point Structures. The experiment to assess the impact of the controller was then repeated with this new controller, with flows of 40, 680, and 630 sccm for the sheath, atomizer, and exhaust, respectively, which provided a line 80 μm wide, and the specification for these lines changed to 80 μm, accordingly. While this output and specification are different than the previous experiment, they should still be accurate for printer behavior and useful. Changes such as this regularly occur as a part of the AJP process, due to unavoidable changes in the ink and flow path of the system. The three sets of ten lines are then printed, and the linewidth and resistance data were analyzed. It was apparent that the controller was working to keep the linewidth at the specification, as shown in Fig. 13, where the controlled average widths were within 3 μm of 80 μm, while the uncontrolled were within 8 μm.

When examining the speed and linewidth of an individual line, such as that shown in Fig. 14, we observe that the PI controller is also able to converge quickly to the specification and does not oscillate. The electrical characterization of the printed lines also shows the effect of the controller, as shown in Fig. 15. While the overall resistance of the line was higher, the variation in resistance with controlled printing is lower, with a range of 0.73 Ω and standard deviation of 0.227 Ω, as compared with 0.96 and 0.314 Ω for uncontrolled. This is desirable, as consistent traces allow for more accurate electrical design.

6 Summary and Conclusions

A closed-loop control of AJP based on image data was implemented. This method of control is based upon the analysis of images captured during the printing process, which provided data on the print, specifically linewidth. Using this data, two types of controllers were implemented, namely a proportional (P) and proportional integral (PI) controller, to vary the print speed in this study.

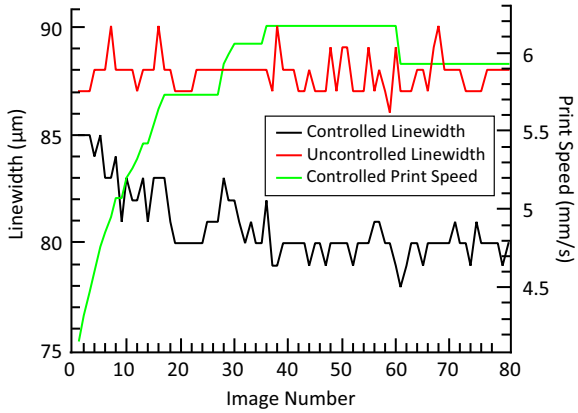

Fig. 14 Plot of linewidths for the second controlled and uncontrolled lines in Set 2, as well as the print speed used for the controlled printing. Note how the linewidth started above specification, but was brought to the specification and maintained there by the controller.

The accuracy of the in situ measurements with in-line measurements was verified with a correlation of 0.99 for the two methods. The linewidth and electrical characteristics of the lines printed under control were also improved, with the linewidths closer to the specification as compared with the uncontrolled prints. The electrical characteristics of the printed lines were enhanced, and the line resistance of the controlled lines was more consistent, albeit with a higher average resistance. This is a crucial enhancement, as it allows practitioners to better design and simulate printed circuits using specified parameters. This enhancement is present for both controllers, but is much improved in the PI controller, where the maximum difference between the average linewidth of a set and the specified linewidth was 1.661 μm versus 5.034 μm and variance among the set averages of 1.548 μm versus 3.205 μm. A similar improvement can be seen in the resistance, where the sets with PI control showed a variance of 0.035 Ω, while P control sets showed a much larger variance in the set averages of 0.1511 Ω. This reduction in geometrical and electrical variance is very important, as it enables the efficient and practical design of circuits, especially at high frequency, where geometrical accuracy has a large influence on key parameters like impedance [2,23–26]. The reduction in the variation shown in these experiments helps move AJP closer to being a manufacturing process, with repeatable, stable parameters for design.

Overall, this is a first attempt at the closed-loop control of a direct-write additive manufacturing process and is mostly an

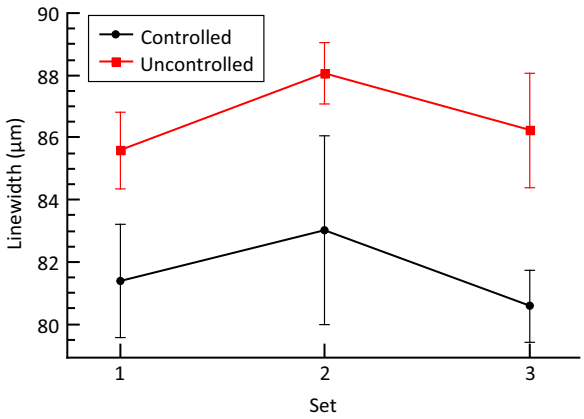

Fig. 13 Plot of controlled and uncontrolled average linewidths using the PI controller. The error bars represent one standard deviation. Note that the specified linewidth of 80 μm is within the error bars of all the controlled sets.

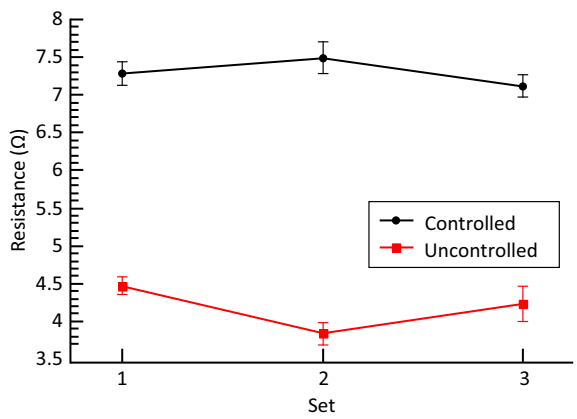

Fig. 15 Plot of the average measured resistance of controlled and uncontrolled lines using the PI controller. Note how the controlled sets are almost in line with each other. Error bars represent one standard deviation.

engineering process control that will eventually be combined with statistical process control [27]. Improvements can also be made to this method by increasing the sample rate, improving image processing, adding capabilities such as the extraction of 3D topology [13,28], and also adopting new control methodologies, such as fuzzy control. Other process parameters, such as the process gas flows, could also be incorporated as control variables, providing the opportunity for multivariate control. Eventually, complex printed structures could be analyzed, as opposed to the simple lines here, and adaptive toolpaths could be generated to compensate for print changes in making complex structures and creating optimal pattern fills, similar to Ref. [29], with active, closed-loop control enhancing the process.

Acknowledgment

This material is based upon work supported, in part, by Air Force Research Laboratory under agreement number FA8650-15-2-5401. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Air Force Research Laboratory or the U.S. Government. The authors would also like to thank S^3IP-CAMM at Binghamton University for their support. One of the authors (PKR) thanks the National Science Foundation for funding his work through Grant Nos. CMMI- 1719388, 1739696, and 1752069. Specifically, the development and application of data analytics and modeling approaches for process modeling and monitoring towards closed-loop control in AM was conceptualized and funded through CMMI-1752069 (CAREER).

References

[1] Zhan, Z., Yu, L., Wei, J., Zheng, C., Sun, D., and Wang, L., 2014, "Application of Aerosol Jet Technology in Through-Via Interconnection for MEMS Wafer-Level Packaging," Microsyst. Technol., **21**(2), pp. 451–455.
[2] Cai, F., Pavlidis, S., Papapolymerou, J., Chang, Y. H., Wang, K., Zhang, C., and Wang, B., 2014, "Aerosol Jet Printing for 3-D Multilayer Passive Microwave Circuitry," 2014 44th European Microwave Conference (EuMC), Rome, Italy, Oct. 6–9, pp. 512–515.
[3] Rudorfer, A., Tscherner, M., Palfinger, C., Reil, F., Hartmann, P., Seferis, I. E., Zych, E., and Wenzl, F. P., 2016, "A Study on Aerosol Jet Printing Technology in LED Module Manufacturing," Fifteenth International Conference on Solid State Lighting and LED-based Illumination Systems, San Diego, CA, Aug. 28– Sept. 1, Proc. SPIE 9954, p. 99540E.
[4] Gupta, A. A., Bolduc, A., Cloutier, S. G., and Izquierdo, R., 2016, "Aerosol Jet Printing for Printed Electronics Rapid Prototyping," 2016 IEEE International Symposium on Circuits and Systems (ISCAS), Montreal, QC, Canada, May 22–25, pp. 866–869.
[5] Stoukatch, S., Laurent, P., Dricot, S., Axisa, F., Seronveaux, L., Vandormael, D., Beeckman, E., Heusdens, B., and Destiné, J., 2012, "Evaluation of Aerosol Jet Printing (AJP) Technology for Electronic Packaging and Interconnect Technique," 2012 4th Electronic System-Integration Technology Conference, Amsterdam, Netherlands, pp. 1–9.
[6] Seifert, T., Baum, M., Roscher, F., Wiemer, M., and Gessner, T., 2015, "Aerosol Jet Printing of Nano Particle Based Electrical Chip Interconnects," Mater. Today: Proc., **2**(8), pp. 4262–4271. nanoFIS 2014 – Functional Integrated nanoSystems.
[7] Aga, R., Lombardi, J., Bartsch, C., and Heckman, E., 2014, "Performance of a Printed Photodetector on a Paper Substrate," IEEE Photonics Technol. Lett., **26**(3), pp. 305–308.
[8] Li, S., Park, J. G., Wang, S., Liang, R., Zhang, C., and Wang, B., 2014, "Working Mechanisms of Strain Sensors Utilizing Aligned Carbon Nanotube Network and Aerosol Jet Printed Electrodes," Carbon, **73**, pp. 303–309.
[9] Paulsen, J. A., Renn, M., Christenson, K., and Plourde, R., 2012, "Printing Conformal Electronics on 3D Structures with Aerosol Jet Technology," Future of Instrumentation International Workshop (FIIW), Gatlinburg, TN, pp. 1–4.
[10] Mahajan, A., Frisbie, C. D., and Francis, L. F., 2013, "Optimization of Aerosol Jet Printing for High-Resolution, High-Aspect Ratio Silver Lines," ACS Appl. Mater. Interfaces, **5**(11), pp. 4856–4864.
[11] Verheecke, W., Van Dyck, M., Vogeler, F., Voet, A., and Valkenaers, H., 2012, "Optimizing Aerosol Jet®Printing of Silver Interconnects on Polyimide Film for Embedded Electronics Applications," 8th International DAAAM Baltic Conference, Tallinn, Estonia, Apr. 19–21, pp. 373–379.
[12] Goth, C., Putzo, S., and Franke, J., 2011, "Aerosol Jet Printing on Rapid Prototyping Materials for Fine Pitch Electronic Applications," 2011 IEEE 61st Electronic Components and Technology Conference (ECTC), Lake Buena Vista, FL, pp. 1211–1216.
[13] Salary, R. R., Lombardi, J. P., Rao, P. K., and Poliks, M. D., 2017, "Online Monitoring of Functional Electrical Properties in Aerosol Jet Printing Additive Manufacturing Process Using Shape-From-Shading Image Analysis," ASME J. Manuf. Sci. Eng., **139**(10), p. 101010.
[14] Salary, R., Lombardi, J., Tootooni, M. S., Donovan, R., Rao, P. K., Borgesen, P., and Poliks, M. D., 2016, "Computational Fluid Dynamics Modeling and Online Monitoring of Aerosol Jet Printing Process," ASME J. Manuf. Sci. Eng., **139**(2), p. 21.
[15] Thompson, B., and Yoon, H.-S., 2015, "Velocity-Regulated Path Planning Algorithm for Aerosol Printing Systems," ASME J. Manuf. Sci. Eng., **137**(3), p. 031020.
[16] Smith, M., Choi, Y. S., Boughey, C., and Kar-Narayan, S., 2017, "Controlling and Assessing the Quality of Aerosol Jet Printed Features for Large Area and Flexible Electronics," Flexible and Printed Electronics, **2**(1), p. 015004.
[17] Gu, Y., Gutierrez, D., Das, S., and Hines, D. R., 2017, "Inkwells for On-Demand Deposition Rate Measurement in Aerosol-Jet Based 3D Printing," J. Micromech. Microeng., **27**(9), p. 097001.
[18] Sun, H., Wang, K., Li, Y., Zhang, C., and Jin, R., 2017, "Quality Modeling of Printed Electronics in aerosol Jet Printing Based on Microscopic Images," ASME J. Manuf. Sci. Eng., **139**(7), p. 071012.
[19] Li, Y., Mohan, K., Sun, H., and Jin, R., 2017, "Ensemble Modeling of In Situ Features for Printed Electronics Manufacturing With In Situ Process Control Potential," IEEE Robot. Autom. Lett., **2**(4), pp. 1864–1870.
[20] Xiong, J., and Zhang, G., Apr. 2014, "Adaptive Control of Deposited Height in GMAW-Based Layer Additive Manufacturing," J. Mater. Process. Technol., **214**(4), pp. 962–968.
[21] Xiong, J., Yin, Z., and Zhang, W., 2016, "Closed-Loop Control of Variable Layer Width for Thin-Walled Parts in Wire and Arc Additive Manufacturing," J. Mater. Process. Technol., **233**, pp. 100–106.
[22] Franklin, G. F., Powell, J. D., and Emami-Naeini, A., 2006, *Feedback Control of Dynamic Systems*, 5th ed, Pearson Prentice Hall, Upper Saddle River, NJ.
[23] Pozar, D. M., 2011, *Microwave Engineering*, 4th ed, Wiley Global Education, Hoboken, NJ.
[24] Cai, F., Chang, Y.-h., Wang, K., Khan, W., Pavlidis, S., and Papapolymerou, J., 2014, "High Resolution Aerosol Jet Printing of D- Band Printed Transmission Lines on Flexible LCP Substrate," 2014 IEEE MTT-S International Microwave Symposium (IMS), Tampa, FL, pp. 1–3.
[25] Cai, F., Chang, Y. H., Wang, K., Zhang, C., Wang, B., and Papapolymerou, J., Oct. 2016, "Low-Loss 3-D Multilayer Transmission Lines and Interconnects Fabricated by Additive Manufacturing Technologies," IEEE Trans. Microw. Theory Techn, **64**(10), pp. 3208–3216.
[26] Godlinski, D., Zichner, R., Zöllmer, V., and Baumann, R. R., 2017, "Printing Technologies for the Manufacturing of Passive Microwave Components: Antennas," Antennas Propag. IET Microw., **11**(14), pp. 2010–2015.
[27] Huang, T., Wang, S., and He, K., 2015, "Quality Control for Fused Deposition Modeling Based Additive Manufacturing: Current research and future trends," 2015 First International Conference on Reliability Systems Engineering (ICRSE), Beijing, China, pp. 1–6.
[28] Pinto-Lopera, J., S. T. Motta, J., and Absi Alfaro, S., 2016, "Real-Time Measurement of Width and Height of Weld Beads in GMAW Processes," Sensors, **16**(9), pp. 1500.
[29] Ding, D., Pan, Z., Cuiuri, D., Li, H., van Duin, S., and Larkin, N., 2016, "Bead Modelling and Implementation of Adaptive MAT Path in Wire and Arc Additive Manufacturing," Robot. Comput.-Integr. Manuf., **39**, pp. 32–42.