

SYSTEMS, METHODS, AND MEDIA FOR MANUFACTURING PROCESSES**Cross-Reference to Related Applications**

[0001] This application claims priority to U.S. Provisional Application Serial No. 63/040,792, filed June 18, 2020, which is hereby incorporated by reference in its entirety. This application is a continuation-in-part of U.S. Provisional Application Serial No. 17/195,746, filed March 9, 2021, which claims priority to U.S. Provisional Application Serial No. 62/986,987 filed March 9, 2020, which is hereby incorporated by reference in its entirety, and is a continuation-in-part of U.S. Application Serial No. 17/091,393, filed November 6, 2020, which claims priority to U.S. Provisional Application Serial No. 62/931,448, filed November 6, 2019, U.S. Provisional Application Serial No. 62/932,063, filed November 7, 2019, and U.S. Provisional Application Serial No. 62/931,453, filed November 6, 2019, which are hereby incorporated by reference in their entireties.

Field of Disclosure

[0002] The present disclosure generally relates to a system, method, and media for manufacturing processes.

Background

[0003] Since the dawn of the industrial revolution in the 18th century, automation has governed the production of goods. Although today's factories have fully embraced automation as a core principle – with robots performing many repeatable tasks in high-production environments – many assembly tasks continue to be performed by humans. These tasks are difficult to automate due to cost, risk of critical failure, or logistics of deploying a robotic system for a low-quantity production run. These production lines are overseen by standard process control and people management, such that an assembler is taught to perform a certain quality metric over time, or they are replaced by another operator. This process has remained largely unchanged since the advent of the assembly.

Summary

[0004] In some embodiments, a manufacturing system is disclosed herein. The manufacturing system includes one or more stations, a monitoring platform, and a control module. Each station is configured to perform at least one step in a multi-step manufacturing process for a component. The monitoring platform configured to monitor progression of the component throughout the multi-step manufacturing process. The control module configured to dynamically adjust processing parameters of a step of the multi-step manufacturing process to

achieve a desired final quality metric for the component, the control module configured to perform operations. The operations include receiving image data of tooling of a first station of the one or more stations. The operations further include identifying a set of keypoints from the image data. The keypoints correspond to position information of the tooling during processing at the first station. The operations further include determining, by a machine learning model, a final quality metric for the component, based on the keypoints. The operations further include, based on the determining, assigning the component to a class of components based on a comparison between the final quality metric generated by the machine learning model and a canonical final quality metric for the component.

[0005] In some embodiments, a computer-implemented method for controlling a multi-step manufacturing process is disclosed herein. The multi-step manufacturing process involves one or more stations of a manufacturing system. Each station is configured to perform at least one step in a multi-step manufacturing process for a component. A computing system associated with the manufacturing system receives image data of tooling of a first station of the one or more stations. The computing system identifies a set of keypoints from the image data, the set of keypoints corresponding to position information of the tooling during processing at the first station. A machine learning model associated with the computing system determines a final quality metric for the component, based on the set of keypoints. Based on the determining, the computing system assigns the component to a class of components based on a comparison between the final quality metric generated by the machine learning model and a canonical final quality metric for the component.

[0006] In some embodiments, a manufacturing system is disclosed herein. The manufacturing system includes one or more stations, a monitoring platform, and a control module. Each station is configured to perform at least one step in a multi-step manufacturing process for a component. The monitoring platform configured to monitor progression of the component throughout the multi-step manufacturing process. The control module configured to dynamically adjust processing parameters of a step of the multi-step manufacturing process to achieve a desired final quality metric for the component, the control module configured to perform operations. The operations include receiving image data of tooling of a first station of the one or more stations. The operations further include identifying a set of keypoints from the image data. The keypoints correspond to position information of the tooling during processing at the first station. The operations further include determining, by a machine learning model, a final quality metric for the component, based on the keypoints. The operations further include

determining that the final quality metric is not within a threshold tolerance from the final quality metric. The operations further include based on the determining, assigning the component to a class of components based on a comparison between the final quality metric generated by the machine learning model and a canonical final quality metric for the component. The operations further include, based on the assigning, determining that the class assigned to the component is not an acceptable class. The operations further include based on the determining, inferring positional information corresponding to the component at the first processing station. The operations further include based on the determining, generating an updated instruction set to be performed by at least one of the first processing station or a downstream station. The operations further include predicting, by a machine learning model, a final quality metric for the component based on the updated instruction set. The operations further include, based on the predicted final quality metric, providing the updated instruction set to at least one of the first processing station or the downstream station.

Brief Description of the Drawings

[0007] So that the manner in which the above recited features of the present disclosure can be understood in detail, a more particular description of the disclosure, briefly summarized above, may be had by reference to embodiments, some of which are illustrated in the appended drawings. It is to be noted, however, that the appended drawings illustrate only typical embodiments of this disclosure and are therefore not to be considered limiting of its scope, for the disclosure may admit to other equally effective embodiments.

[0008] Figure 1 is a block diagram illustrating a manufacturing environment, according to example embodiments.

[0009] Figure 2 is a block diagram illustrating control module, according to exemplary embodiments.

[0010] Figure 3 is a block diagram illustrating an exemplary architecture of LSTM model, according to example embodiments.

[0011] Figure 4 is a block diagram visually illustrating the overall process flow of a feedback segment for tooling module, according to example embodiments.

[0012] Figure 5 is a block diagram illustrating architecture of GRU model, according to example embodiments.

[0013] Figure 6A is a flow diagram illustrating a method of correcting a multi-step manufacturing process, according to example embodiments.

[0014] Figure 6B is a flow diagram illustrating a method of correcting a multi-step manufacturing process, according to example embodiments.

[0015] Figure 7 is a flow diagram illustrating a method of correcting a multi-step manufacturing process, according to example embodiments.

[0016] Figure 8A illustrates a system bus computing system architecture, according to example embodiments.

[0017] Figure 8B illustrates a computer system having a chipset architecture, according to example embodiments.

[0018] To facilitate understanding, identical reference numerals have been used, where possible, to designate identical elements that are common to the figures. It is contemplated that elements disclosed in one embodiment may be beneficially utilized on other embodiments without specific recitation.

Detailed Description

[0019] Manufacturing processes may be complex and include raw materials being processed by different process stations (or “stations”) until a final component is produced. In some embodiments, each process station receives an input for processing and may output an intermediate output that may be passed along to a subsequent (downstream) process station for additional processing. In some embodiments, a final process station may receive an input for processing and may output the final component or, more generally, the final output.

[0020] In some embodiments, each station may include one or more tools/equipment that may perform a set of process steps. Exemplary process stations may include, but are not limited to, conveyor belts, injection molding presses, cutting machines, die stamping machines, extruders, computer numerical control (CNC) mills, grinders, assembly stations, three-dimensional printers, quality control stations, validation stations, and the like.

[0021] In some embodiments, operations of each process station may be governed by one or more process controllers. In some embodiments, each process station may include one or more process controllers that may be programmed to control the operation of the process station. In some embodiments, an operator, or control algorithms, may provide the station controller with station controller setpoints that may represent the desired value, or range of values, for each control value. In some embodiments, values used for feedback or feed forward in a manufacturing process may be referred to as control values. Exemplary control values may include, but are not limited to: speed, temperature, pressure, vacuum, rotation, current, voltage,

power, viscosity, materials/resources used at the station, throughput rate, outage time, noxious fumes, and the like.

[0022] One or more techniques provided herein improves upon conventional processes by providing a system through which image and/or video data may be used to predict or forecast a final quality metric of a component. In some embodiments, the system may include a monitoring system configured to capture or record video and/or image data of the tooling of each processing node or station of manufacturing system. Based on the positioning of the tooling during a process step, the present system may be configured to predict or forecast a final quality metric of a component. If the predicted final quality metric falls outside of a range of acceptable values, the present system may generate and provide updated processing instructions to at least one of the current processing nodes k or stations k or to processing nodes $(k + 1)$ and/or stations $(k + 1)$, in an attempt to correct for any errors in processing, such that a desired final quality metric may be achieved.

[0023] In some embodiments, the system may include a monitoring system configured to capture or record video and/or image data of a component at each processing node or station of the manufacturing system. Based on visual information about the component at an end of a processing node, the present system may be configured to predict or forecast a final quality metric of a component. If the predicted final quality metric falls outside of a range of acceptable values, the present system may generate and provide updated processing instructions to at least one of processing nodes k or stations k or to processing nodes $(k + 1)$ and/or stations $(k + 1)$, in an attempt to correct for any errors in processing, such that a desired final quality metric may be achieved.

[0024] In this manner, the present system is able to predict or forecast a final quality metric of a component, at any stage of the manufacturing process, without having to actually test the component. Such system is particularly useful for final quality metrics that would otherwise require destructive testing, or for which the final quality metric cannot be evaluated until processing is complete.

[0025] Figure 1 is a block diagram illustrating a manufacturing environment 100, according to example embodiments. Manufacturing environment 100 may include a manufacturing system 102, a monitoring platform 104, and a control module 106. Manufacturing system 102 may be broadly representative of a multi-step manufacturing system. In some embodiments, manufacturing system 102 may be representative of an assembly line system, where each processing station may be representative of a human worker. In some embodiments,

manufacturing system 102 may be representative of a manufacturing system for use in additive manufacturing (e.g., 3D printing system). In some embodiments, manufacturing system 102 may be representative of a manufacturing system for use in subtractive manufacturing (e.g., CNC machining). In some embodiments, manufacturing system 102 may be representative of a manufacturing system for use in a combination of additive manufacturing and subtractive manufacturing. More generally, in some embodiments, manufacturing system 102 may be representative of a manufacturing system for use in a general manufacturing process.

[0026] Manufacturing system 102 may include one or more stations 108₁-108_n (generally, “station 108”). Each station 108 may be representative of a step and/or station in a multi-step manufacturing process. For example, each station 108 may be representative of a layer deposition operation in a 3D printing process (e.g., station 108₁ may correspond to layer 1, station 108₂ may correspond to layer 2, etc.). In another example, each station 108 may correspond to a specific processing station. In another example, each station 108 may correspond to a specific human operator performing a specific task in an assembly line manufacturing process.

[0027] Each station 108 may include a process controller 114 and control logic 116. Each process controller 114₁-114_n may be programmed to control the operation of each respective station 108. In some embodiments, control module 106 may provide each process controller 114 with station controller setpoints that may represent the desired value, or range of values, for each control value. Control logic 116 may refer to the attributes/parameters associated with a station’s 108 process steps. In operation, control logic 116 for each station 108 may be dynamically updated throughout the manufacturing process by control module 106, depending on a current trajectory of a final quality metric.

[0028] Monitoring platform 104 may be configured to monitor each station 108 of manufacturing system 102. In some embodiments, monitoring platform 104 may be a component of manufacturing system 102. For example, monitoring platform 104 may be a component of a 3D printing system. In some embodiments, monitoring platform 104 may be independent of manufacturing system 102. For example, monitoring platform 104 may be retrofit onto an existing manufacturing system 102. In some embodiments, monitoring platform 104 may be representative of an imaging device configured to capture an image of a component or tooling (e.g., a worker or a process tool) at each step of a multi-step process. For example, monitoring platform 104 may be configured to capture an image of the component at each station 108 and/or an image of a component developing the component at each station

108 (e.g., tooling, human, etc.). Generally, monitoring platform 104 may be configured to capture information associated with production of a component (e.g., an image, a voltage reading, a speed reading, etc.) and/or tool (e.g., hand position, tooling position, etc.), and provide that information, as input, to control module 106 for evaluation.

[0029] Control module 106 may be in communication with manufacturing system 102 and monitoring platform 104 via one or more communication channels. In some embodiments, the one or more communication channels may be representative of individual connections via the Internet, such as cellular or Wi-Fi networks. In some embodiments, the one or more communication channels may connect terminals, services, and mobile devices using direct connections, such as radio frequency identification (RFID), near-field communication (NFC), Bluetooth™, low-energy Bluetooth™ (BLE), Wi-Fi™, ZigBee™, ambient backscatter communication (ABC) protocols, USB, WAN, or LAN.

[0030] Control module 106 may be configured to control each process controller of manufacturing system 102. For example, based on information captured by monitoring platform 104, control module 106 may be configured to adjust process controls associated with a specific station 108. In some embodiments, control module 106 may be configured to adjust process controls of a specific station 108 based on a projected final quality metric.

[0031] Figure 2 is a block diagram illustrating control module 106, according to exemplary embodiments. Control module 106 may include a tooling module 202 and a component module 204.

[0032] Tooling module 202 may be configured to project a final quality metric of a specimen at a given stage of production based on image data obtained by monitoring platform 104. In operation, control module 106 may receive input from monitoring platform 104. In some embodiments, such input may take the form of an image or video of the tooling performing a subprocess at a given step of the multi-step manufacturing process. For example, the image or video data may include image or video data of a human's hands while performing a specific subprocess of the multi-step manufacturing process. In another example, the image or video data may include image or video data of a three-dimensional printer performing depositing a specific layer of a multi-layer manufacturing process. Based on the input, control module 106 may project a final quality metric of the component. Depending on the projected final quality metric of the component, control module 106 may determine one or more actions to take in subsequent manufacturing steps in order to reach a desired or threshold final quality metric. For example, if the projected final quality metric falls outside of a range of acceptable values,

control module 106 may take one or more actions to rectify the manufacturing process. In some embodiments, control module 106 may interface with station controllers in at least one of a current station $108(k)$ or subsequent stations $108(k + 1)$ to adjust their respective control and/or station parameters. In some embodiments, control module 106 may provide human manufacturers with updated instructions to be performed at each processing station $(k + 1)$ of a production line. These adjustments may aid in correcting the manufacturing process, such that the final quality metric may be within the range of acceptable quality metrics.

[0033] Component module 204 may be configured to project a final quality metric of a specimen at a given stage of production based on image data obtained by monitoring platform 104. In some embodiments, component module 204 may receive input from monitoring platform 104. In some embodiments, such input may take the form of an image or video of the component at a given step of the multi-step manufacturing process. In some embodiments, component module 204 may receive inferred component data at a given step of the multi-step manufacturing process from tooling module 202. For example, tooling module 202 may infer information about the component at a given step of the multi-step manufacturing process based on the tooling image or video data and provide that inferred information to component module 204 as input. Based on the input, component module 204 may project a final quality metric of the component. Depending on the projected final quality metric of the component, component module 204 may determine one or more actions to take in subsequent manufacturing steps in order to reach a desired or threshold final quality metric. For example, if the projected final quality metric falls outside of a range of acceptable values, component module 204 may identify one or more actions to rectify the manufacturing process. In some embodiments, control module 106 may interface with station controllers in at least one of the current station $108k$ or subsequent stations $108(k + 1)$ to adjust their respective control and/or station parameters. In some embodiments, control module 106 may provide human manufacturers with updated instructions to be performed in at least one of a current processing station k or other processing stations $(k + 1)$ of a production line. These adjustments may aid in correcting the manufacturing process, such that the final quality metric may be within the range of acceptable quality metrics.

[0034] Each of tooling module 202 and component module 204 may include one or more software modules. The one or more software modules may be collections of code or instructions stored on a media (e.g., memory of computing systems associated with control module 106) that represent a series of machine instructions (e.g., program code) that

implements one or more algorithmic steps. Such machine instructions may be the actual computer code the processor interprets to implement the instructions or, alternatively, may be a higher level of coding of the instructions that is interpreted to obtain the actual computer code. The one or more software modules may also include one or more hardware components. One or more aspects of an example algorithm may be performed by the hardware components (e.g., circuitry) itself, rather as a result of the instructions. Further, in some embodiments, each of tooling module 202 and component module 204 may be configured to transmit one or more signals among the components. In such embodiments, such signals may not be limited to machine instructions executed by a computing device.

[0035] In some embodiments, tooling module 202 and component module 204 may communicate via one or more local networks. Network may be of any suitable type, including individual connections via the Internet, such as cellular or Wi-Fi networks. In some embodiments, network may connect terminals, services, and mobile devices using direct connections, such as radio frequency identification (RFID), near-field communication (NFC), Bluetooth™, low-energy Bluetooth™ (BLE), Wi-Fi™, ZigBee™, ambient backscatter communication (ABC) protocols, USB, WAN, or LAN. Because the information transmitted may be personal or confidential, security concerns may dictate one or more of these types of connection be encrypted or otherwise secured. In some embodiments, however, the information being transmitted may be less personal, and therefore, the network connections may be selected for convenience over security.

[0036] Tooling module 202 may include acquisition system 206, extractor module 208, and prediction module 210. Generally, a multi-node or multi-station assembly environment, such as manufacturing system 102, may be represented, broadly as $G(\vec{s}_i, \vec{a}_i)$, where $\vec{s}_i$ may represent the states of the component at all i nodes and where $\vec{a}_i$ may represent that set of actions to be performed on the component by the tooling at all i nodes. Given a norm or canonical quality measurement, V_c , tooling module 202 may be configured to optimize the error in the assembly process with an estimated quality metric, $\hat{V}_c$, such that $\hat{V}_c$ may be within a threshold distance of V_c . In some embodiments, tooling module 202 may estimate the state, $\vec{s} = [s_0, s_1, \dots, s_{N-1}]$, which may be a numerical representation of a state of the component at all N nodes, and the actions, $\vec{a} = [a_0, a_1, \dots, a_{N-1}]$, which may represent the instructions or control values at each node.

[0037] Acquisition system 206 may be configured to receive image data of the assembly process at each node, N . In some embodiments, acquisition system 206 may receive the image

data of the assembly process of monitoring platform 104. In some embodiments, for each node N , acquisition system 206 may receive V number of images, where V may represent the number of cameras of monitoring platform 104 that may record the assembly procedure at each node. Accordingly, each image of the V number of images may capture a different perspective of the component during processing. Following receipt of the image data, acquisition system 206 may be configured to extract a subset of images of frames. For example, acquisition system 206 may be configured to extract L number of images from the received image data. The extracted images may be referred to as landmark frames. Landmark frames may be those image frames that are in high motion. The extracted images may include those images or frames that include certain landmarks, $\vec{I}$, of the component, where $\vec{I} = [i_0, i_1, \dots, i_{L-1}]$, and may represent the entire manufacturing process for the component.

[0038] Both minimal-motion (e.g., “landmark”) frames and maximal-motion (e.g., “high motion”) frames for a given operator may contain useful information for a classifier that is trying to correlate finger-hand-arm data and flight performance data in a robust way across a number of operators. In some embodiments, an optical flow algorithm may be used to measure an amount of motion in any given frame. Acquisition system 206 may select those frames that contain the most motion.

[0039] Extractor module 208 may be configured to extract keypoints from the L number of images. For example, extractor module 208 may be configured to extract K number of keypoints, i.e., (x, y) pairs, per landmark I_l . In other words, extractor module 208 may be configured to output K number of keypoints for a given input, $I_l \in \mathbb{R}^{w \times h}$, where $l \in [0, L - 1]$. As output, extractor module 208 may generate a single vector $\vec{P} = [\vec{p}_0, \vec{p}_1, \dots, \vec{p}_{L-1}]$. This vector may include landmark representations of K number of (x, y) pairs which may be represented by $\vec{p}_i = [(x_0, y_0), (x_1, y_1), \dots, (x_{K-1}, y_{K-1})]$, where $i \in [0, L - 1]$.

[0040] In some embodiments, to generate the $\vec{P}$, extractor module 208 may implement two separate algorithms: (1) a bounding box estimation; and (2) keypoint detection.

[0041] With respect to the bounding box estimation, given $\vec{I}$, each landmark frame may be processed with a threshold image segmentation to generate a mask image for each tooling component. For example, in an embodiment in which the tooling is a human, extractor module 208 may generate a mask image for each of the user’s hands. In some embodiments, extractor module 208 may implement blob detection to locate components of the tooling. Using a human as an example, extractor module 208 may assume that the image always contains both the left

and right hands of the user. When a frame fails to include both hands, extractor module 208 may assign the value with an arbitrary constant value, c .

[0042] With respect to keypoint detection, extractor module 208 may identify keypoints of the tooling based on the estimated bounding boxes. For example, with given input, extractor module 208 may estimate K number of points, $\vec{p}_i$, along with its confidence value, $\vec{c}_i$. In some embodiments, extractor module 208 may estimate, not only the points that are visible on the frame, but also points that may be occluded from the frame due to one or more of articulation, viewpoints, objects, or tool interactions. Because the objective may be to predict the quality measurement using the tracked keypoints, the non-estimated occluded points may be unique and important features representing the assembly process. Therefore, an occlusion threshold value, t_o , may be derived from observation of c_i value on occluded points in small randomly chosen subset of the landmark frames. Using t_o , extractor module 208 may filter out the estimation that are $c_o < t_o$. For those filtered points, extractor module 208 may assign them an arbitrary constant value, c . Regardless of the visibility of the tooling or keypoints on the frame, the output of extractor module 208 may include $L * K(x, y)$ pairs for each component.

[0043] In other words, extractor module 208 may be configured to assign a default basal level of confidence for occluded components and, therefore, estimate those keypoints so that a full set of keypoints may be available to measure discrepancies between observed trial tooling component positions and the canonical tooling component positions. In some embodiments, tooling module 202 may draw inferences about the state of the component at a given point in time (e.g., station i) and then output recommended modified-from-canonical subsequent actions to take to correct for the measured discrepancies.

[0044] Prediction module 210 may be configured to predict a final quality metric, $\hat{V}_c$. For example, prediction module 210 may be configured to predict a final quality metric, $\hat{V}_c$, where $\hat{V}_c \in \mathbb{R}^1$, given tooling tracking information for the L number of points-in-time, $\vec{P}$, where $\vec{P} \in \mathbb{R}^{K*L}$ may be gathered from each processing station 108. In some embodiments, prediction module 210 may implement a long short-term memory (LSTM) model to output the final quality metric. The LSTM model may allow prediction module 210 to overcome a vanishing gradient problem that is common in conventional recurrent neural networks. The vanishing gradient problem is the case where a model fails to relate the early weights of the neural network when the magnitude of the gradients are small for later layers. LSTM model eliminates this issue.

[0045] In some embodiments, tooling module 202 may further include a classification module 215. Classification module 215 may be configured to classify a specimen into one or more classes based on the predicted final quality metric, $\hat{V}_c$. For example, in operation, classification module 215 may receive a canonical or desired final quality metric, V_c , for the specimen. Classification module 215 may be configured to compare the predicted final quality metric, $\hat{V}_c$, to the canonical or desired final quality metric, V_c , to generate a delta, Δ_V . Depending on the value of Δ_V , classification module 215 may sort the specimen into one or more classes. For example, assume that classification module 215 may sort the specimen into one of four classes: fail (class 1), fair (class 2), excellent (class 3), and over-qualify (class 4). Classification module 215 may sort the specimen based on associated class definitions. For example, the associated definitions may be:

$$\text{Class 1 : Fail : } \hat{V}_c < \frac{V_c}{3}$$

$$\text{Class 2 : Fair : } \frac{V_c}{3} \leq \hat{V}_c < \frac{2V_c}{3}$$

$$\text{Class 3 : Excellent : } \frac{2V_c}{3} \leq \hat{V}_c \leq V_c$$

$$\text{Class 4 : Over Qualify : } V_c < \hat{V}_c$$

[0046] As those skilled in the art recognize, the associated definitions may change based on operator or client preference.

[0047] In some embodiments, control module 106 may use the classification assigned to the component to determine whether the final quality metric is acceptable. For example, an operator or client may specify that components classified in Class 1 are not acceptable, while components classified in any of Classes 2-4 are acceptable. In another example, an operator or client may specify that components classified in Class 1 or Class 2 are not acceptable, while components classified in Class 3 or Class 4 are acceptable.

[0048] Figure 3 is a block diagram illustrating an exemplary architecture of LSTM model, according to example embodiments. As shown, LSTM model 300 may include three layers 302₁-302₃ (generally layer 302). As shown, each layer 302 may include one or more cells 304. In some embodiment, each cell's 304 input may be p_i , where $i \in [0, L - 1]$. Each cell in the LSTM may be defined as:

$$\text{eq. 1: } f_i = \sigma(W_f * [h_{i-1}, p_i] + b_f)$$

$$\text{eq. 2: } I_i = \sigma(W_I * [h_{i-1}, p_i] + b_I)$$

$$eq.3: c = f_i * c_{t-1} + I_i * \tanh(W_c * [h_{t-1}, p_i] + b_c)$$

$$eq.4: o_i = \sigma(W_o * [h_{t-1}, p_i] + b_o)$$

$$eq.5: h_i = o_i * \tanh c_i$$

where equation 1 decides whether to keep information from the previous cell or not; equation (2) decides which values to update; equation (3) updates the cell; and equation (4) decides which part to output. Equation (5) may filter the output parts so that LSTM model 300 only outputs what it is programmed to output.

[0049] In some embodiments, LSTM model 300 may include three layers 302 with 30 hidden sizes. In some embodiments, LSTM model 300 may be a sequence-to-one LSTM model. For training, the L1, mean absolute error (MAE), loss function:

$$\mathcal{L}(\hat{y}, \tilde{y}) = \frac{\sum_{b=1}^B |y_b - \hat{y}_b|}{B}$$

may be minimized using an Adam optimizer. In some embodiments, MAE may be used because the goal may be to minimize or reduce the magnitude of errors, regardless of the direction of the error.

[0050] Referring back to Figure 2, prediction module 210 may be trained using a canonical set of instructions, $\vec{a}$, and multiple components, M . For example, using 10 node videos of M number of data, the input for prediction module 210 may be structured by first being pre-processed through acquisition system 206 and extractor module 208. In some embodiments, each component involved in training may be validated through a validation algorithm that checks their composition (e.g., shape). In some embodiments, the validation algorithm may compute a similarity index on the composition at the end of each node by comparing the composition with canonical compositions. As a result, the components for training are roughly similar to the canonical composition.

[0051] In some embodiments, for output, the corresponding physical components may be tested in a controlled environment to measure their quality metric. Using the prepared input data with corresponding output data, prediction module 210 may be trained, for example, with a portion of M being training data and another portion of M being validation data. Once trained, prediction module 210 may be able to predict a quality measurement of a component at a given processing step, based on image data of the tooling.

[0052] Now referring to component module 204, component module 204 may include stochastic gradient descent (SGD) module 212, gated recurrent unit (GRU) model 214, and

simulation module 216. For purposes of this discussion, a partial construct of a component may be defined as $\hat{a}_k$, where step k introduces an irreversible error in the manufacturing process, and steps $k + 1, \dots, N$ are not yet defined. Component module 204 may be configured to identify an optimal corrective sequence of remaining actions $[\{r_{k+1}, t_{k+1}\}, \dots, \{r_N, t_N\}]$, where r and t may correspond to specific operations to be performed on the component, at least one of the current processing station or each subsequent processing station ($k + 1$ to N) of manufacturing system 102. More generally, any component $\vec{a}$ may be defined as a sequence of all operations performed at each processing station $1 \dots N$ of manufacturing system. Mathematically, $\vec{a} = [\{r_1, t_1\}, \dots, \{r_N, t_N\}]$. At each manufacturing step $i = 1, \dots, 10$, the virtual representation system may represent the component in Euclidean space (e.g., $\mathbb{R}^3$) as a set of connected surface and a set of connected points uniformly distributed along the outer contour of each surface. In some embodiments, the virtual representation functions generating these representations may be referred to as $S(\vec{a}_i)$ and $P(\vec{a}_i)$, respectively. In some embodiments, component module 204 may be configured to correct $\vec{a}_k$ toward a particular, canonical component, $\vec{a}^*$.

[0053] Simulation module 216 may be configured to simulate or generate a surface model for a given component $\hat{a}_i$. For example, simulation module 216 may receive tooling information from tooling module 202. Based on the keypoints generated by tooling module 202, simulation module 216 may be configured to generate a surface model representing a state of the component $\hat{a}_i$ at a specific process step i . In some embodiments, the surface model may be represented as $S(\vec{a}_i)$. In some embodiments, simulation module 216 may further be configured to generate or estimate a quality metric of the component $\vec{a}_i$. From the surface model, simulation module 216 may be configured to generate a points model, $P(\vec{a}_i)$, representing specific coordinates of the component, $\vec{a}_i$. For example, from surface model $S(\vec{a}_i)$, simulation module 216 may create points model, $P(\vec{a}_i)$, by placing a number of points uniformly spaced around a bounding contour of each surface in $S(\vec{a}_i)$. In some embodiments, $S(\vec{a}_i)$ may be used to simulate performance of the artifact $\vec{a}_i$.

[0054] SGD module 212 may receive the points model, $P(\vec{a}_i)$, from simulation module 216. SGD module 212 may determine whether an irreversible error, k , has occurred by comparing the points model, $P(\vec{a}_i)$, at step i to a canonical points model, $P(\vec{a}^*)$ of a canonical component, $\vec{a}^*$. An irreversible error may be defined to be a measurably significant structural deviation from a canonical component at step k . SGD module 212 may be configured to detect an irreversible error by taking a Hausdorff distance. For example, SGD module 212 may match

a processing step of current component $\vec{a}$ to canonical component $\vec{a}^*$ based on respective Euclidean point sets. Mathematically, SGD module 212 may be configured to compute the Hausdorff distance between $P(\vec{a}_i)$ and $P(\vec{a}^*)$ for and $i \in 1, \dots, N$. For example,

$$h(X, Y) = \max_{x \in X} \{\min_{y \in Y} \{d(x, y)\}\}$$

where $d(x, y)$ may be the Euclidean distance between x and y , and the undirected Hausdorff distance may be:

$$H(X, Y) = \max \{h(X, Y), h(Y, X)\}$$

[0055] An irreversible error may be present when the Hausdorff distance between the current component $\vec{a}_i$ and the canonical component $\vec{a}^*$ exceeds some threshold tolerance. For example, SGD module 212 may determine that an error occurs at step k when:

$$H(P(\vec{a}_k), P(\vec{a}^*)) > \tau_H$$

where τ_H is some suitably defined tolerance threshold.

[0056] Assuming an error is present, SGD module 212 may be configured to construct a set of updated actions $\{[r_{k+1}, t_{k+1}], \dots, [r_N, t_N]\}$ given a set of actions up to a point of error an irreversible error k . In some embodiments, this set of updated actions may be referred to as x_{tail} . The sequences of steps or actions that preceded and included the error step k may be referred to as x_{head} . Together, x_{tail} and x_{head} may define a component $\vec{a}_i$. Based on x_{head} , SGD module 212 may solve for x_{tail} using a stochastic gradient descent method.

[0057] GRU model 214 may be configured to predict a final quality metric for the component, $\vec{a}_i$, based on $x_{tail} \oplus x_{head}$, where $\oplus$ may represent a vector concatenation operator. The final quality metric generated by GRU model 214 may be compared against a canonical final quality metric to determine if x_{tail} is proper. For example, assuming that the combination of x_{tail} and x_{head} yields a final quality metric that falls outside of a range of acceptable values, GRU model 214 may instruct SGD module 212 to generate an updated sequence of updated actions for further evaluation.

[0058] Figure 5 is a block diagram illustrating architecture of GRU model 214, according to example embodiments. As shown, GRU model 214 may include N GRU cells 502₁-502_N (generally, GRU cell 502), with each GRU cell 502 corresponding to a respective processing station 108. Each GRU cell 502 may include an (r_i, t_i) input pair and a hidden state output $\vec{h}_i$ of a predetermined size. Together, these input pairs (r_i, t_i) may define a given component $\vec{a}_i$. In some embodiments, each GRU cell 502 may be defined by:

$$\begin{aligned}
\vec{r}_t &= \sigma(W_{ir}\vec{x}_t + b_{ir} + W_{hr}\vec{h}_{t-1} + \vec{b}_{hr}) \\
\vec{z}_t &= \sigma(W_{iz}\vec{x}_t + \vec{b}_{iz} + W_{hz}\vec{h}_{t-1} + \vec{b}_{hz}) \\
\vec{n}_t &= \tanh(W_{in}\vec{x}_t + \vec{b}_{in} + \vec{r}_t(W_{hn}\vec{h}_{t-1} + \vec{b}_{hn})) \\
\vec{h}_t &= (\vec{1} - \vec{z}_t)\vec{n}_t + \vec{z}_t\vec{h}_{t-1}
\end{aligned}$$

where $\vec{h}_t$ may be the hidden state at time t , $\vec{x}_t$ may be the input at time t , and $\vec{r}_t$, $\vec{z}_t$, and $\vec{n}_t$ may represent the reset, update, and new gates at time t , respectively.

[0059] GRU model 214 may be trained to generate weights corresponding thereto. For example, GRU model 214 may be trained iteratively to bias GRU model 214 toward solving a specific subproblem. In some embodiments, during a first iteration, GRU model 214 may generate a plurality (e.g., several hundred to several thousand) best predictions (one for each possible error at step k) to complete a given component $\vec{a}_k$ at steps $k + 1, \dots, N$, along with the corresponding predicted quality metrics corresponding to each prediction.

[0060] In some embodiments, all of the predicted completions may be rendered in a virtual representation system, their stepwise Hausdorff distances computed, and their rendered surfaces simulated, to obtain distance measures between the generated predictions and the canonical. In some embodiments, the loss values between the canonical and predicted quality metric measures may be computed and fed back into the GRU model 214, whose weights may be adjusted via backpropagation, producing a second iteration. This process may continue until a desired error threshold is obtained.

[0061] In some embodiments, GRU model 214 may be trained on a prepared or synthetic data set. For example, simulator module 216 or another simulation system (e.g., a CAD model) may be used to generate a training set for training of GRU model 214. For example, a simulation system may generate, for example, a set of prepared components (e.g., 100,000 prepared components) to represent many unbiased randomized trials of actions that may cover the full admissible range of all parameter values. In some embodiments, the set prepared components may also include biased randomized trials of action, in addition to the unbiased randomized trials. This set may be referred to as $P(U(0,1), 100000)$. Each $\vec{p} \in P(U(0,1), 100000)$ may obtain its defining parameter values by sampling from the uniform distribution $U(0,1)$.

[0062] In some embodiments, for example, a set of 50,000 prepared components may be generated to represent many trials of processing a component to replicate the canonical

component $\vec{a}^*$. This may be referred to as the set $P(N(\vec{a}^*), 50000)$. Each $p \in P(N(\vec{a}^*), 50000)$ may obtain its defining parameter values by a statistical sampling process. In some embodiments, the i^{th} action may centrally tend to the corresponding canonically-defined fold and may vary according to an appropriately parameterized Gaussian distribution. The prepared component data set may be defined as $P = P(U(0,1), 100000) \cup P(N(\vec{a}^*), 50000)$.

[0063] Figure 4 is a block diagram visually illustrating the overall process flow of a feedback segment 400 for tooling module 202, according to example embodiments.

[0064] As shown, feedback segment 400 may include a first portion 402, a second portion 404, and a third portion 406. During first portion 402, tooling module 202 may perform an acquisition process, such as that carried out, at least partially, by acquisition system 206.

[0065] As shown, at block 408, manufacturing system 102 may receive processing instructions for processing a component. In some embodiments, manufacturing system 102 may receive processing instructions on a per station 108 basis. For example, each station 108 of manufacturing system 102 may receive independent processing instructions. In some embodiments, processing instructions may include control values that define attributes (e.g., temperature, pressure, etc.) of a station 108 for manufacturing. In some embodiments, processing instructions may include videos or images that visually illustrate to a human operator how to perform a specific processing step at a processing node of the manufacturing process.

[0066] At block 410, acquisition system 206 may receive image data from monitoring platform 104. In some embodiments, for each node N, acquisition system 206 may receive V number of images, where V may represent the number of cameras of monitoring platform 104 that may record the assembly procedure at each node. Following receipt of the image data, at block 412, acquisition system 206 may be configured to extract a subset of images of frames. For example, acquisition system 206 may be configured to extract L number of images from the received image data. The extracted images may be referred to as landmark frames. The extracted images may include those images or frames that include certain landmarks and may represent the entire manufacturing process for the component.

[0067] Second portion 404 may correspond to operations performed by extractor module 208. As shown, extractor module 208 may receive at least the extracted images from acquisition system 206. Extractor module 208 may be configured to extract keypoints from the L number

of images. For example, extractor module 208 may be configured to extract K number of keypoints, i.e., (x, y) pairs, per landmark I_i .

[0068] At block 414, extractor module 208 may perform bounding box estimation. For example, given $\vec{I}$, each landmark frame may be processed with a threshold image segmentation to generate a mask image for each tooling component. For example, as shown, in an embodiment in which the tooling is a human, extractor module 208 may generate a mask image for each of the user's hands.

[0069] At block 416, extractor module 208 may perform keypoint detection, given the bounding box estimation. For example, with given input, extractor module 208 may estimate K number of points, $\vec{p}_i$, along with its confidence value, $\vec{c}_i$. In some embodiments, extractor module 208 may estimate, not only the points that are visible on the frame, but also points that may be occluded from the frame due to one or more of articulation, viewpoints, objects, or tool interactions.

[0070] Third portion 406 may correspond to operations performed by prediction module 210. As shown, at block 418, prediction module 210 may receive keypoint information from extractor module 208 and may be configured to predict a final quality metric, $\hat{V}_c$. In some embodiments, prediction module 210 may implement a long short-term memory (LSTM) model to output the final quality metric.

[0071] Figure 6A is a flow diagram illustrating a method 600 of correcting a multi-step manufacturing process, according to example embodiments. Method 600 may begin at step 602.

[0072] At step 602, an instruction set may be provided to manufacturing system 102. The instruction set may be representative of a set of instructions for a manufacturing process to be carried out by manufacturing system 102. In some embodiments, the instruction set may be provided to each station 108. For example, each canonical instruction set provided to each respective station 108 may define the processing parameters for a specific manufacturing step. In another example, each canonical instruction set may be a video of discrete steps to be performed by a human actor at a specific processing node or station 108.

[0073] At step 604, control module 106 may receive image data of the tooling (e.g., station 108) from monitoring platform 104. For example, acquisition system 206 may receive image data of the assembly process at a respective processing node. In some embodiments, acquisition system 206 may receive V number of images, where V may represent the number of cameras of monitoring platform 104 that may record the assembly procedure at a specific

procession station 108. Accordingly, each image of the V number of images may capture a different perspective of the tooling during processing.

[0074] At step 606, control module 106 may extract a subset of images from the obtained image data. For example, following receipt of the image data, acquisition system 206 may be configured to extract a subset of images of frames. For example, acquisition system 206 may be configured to extract L number of images, i.e., landmark frames, from the received image data. The extracted images may be referred to as landmark frames. Landmark frames may be those image frames that are in high motion. The extracted images may include those images or frames that include certain landmarks, $\vec{l}$, of the component, where $\vec{l} = [l_0, l_1, \dots, l_{L-1}]$, and may represent the entire manufacturing process for the component.

[0075] At step 608, control module 106 may extract one or more keypoints of the tooling from the landmark frames. For example, extractor module 208 may extract keypoints from the L number of images. Extractor module 208 may identify or extract K number of keypoints for a given input, $I_l \in \mathbb{R}^{w \times h}$, where $l \in [0, L - 1]$. As output, extractor module 208 may generate a single vector $\vec{P} = [\vec{p}_0, \vec{p}_1, \dots, \vec{p}_{L-1}]$. This vector may include landmark representations of K number of (x, y) pairs which may be represented by $\vec{p}_i = [(x_0, y_0), (x_1, y_1), \dots, (x_{K-1}, y_{K-1})]$, where $i \in [0, L - 1]$.

[0076] At step 610, control module 106 may predict a final quality metric for the component based on at least the identified keypoints. In some embodiments, prediction module 210 may implement a long short-term memory (LSTM) model to output the final quality metric.

[0077] At step 612, control module 106 may compare the final quality metric to a desired quality metric. If, at step 612, control module 106 determines that the final quality metric is a threshold tolerance of the desired quality metric, the manufacturing process may proceed to the next processing station or node (e.g., step 616), in accordance with the original instruction set. If, however, at step 612, control module 106 determines that the final quality metric is not within a threshold tolerance of the desired quality metric, then, at step 614 control module 106 may adjust the processing parameters of at least one of the processing station k or one or more processing stations $k + 1$, where k may represent the current processing station and $k+1$ may represent other processing stations downstream of the current processing station. In some embodiments, control module 106 may interface with station controllers in a current station 108(k) and/or subsequent stations 108 ($k + 1$) to adjust their respective control and/or station parameters. In some embodiments, control module 106 may provide human manufacturers with updated instructions to be performed in at least one of a processing station k or one or

more processing station $(k + 1)$ of a production line. These adjustments may aid in correcting the manufacturing process, such that the final quality metric may be within the range of acceptable quality metrics.

[0078] Figure 6B is a flow diagram illustrating a method 650 of correcting a multi-step manufacturing process, according to example embodiments. Method 650 may begin at step 652.

[0079] At step 652, an instruction set may be provided to manufacturing system 102. The instruction set may be representative of a set of instructions for a manufacturing process to be carried out by manufacturing system 102. In some embodiments, the instruction set may be provided to each station 108. For example, each canonical instruction set provided to each respective station 108 may define the processing parameters for a specific manufacturing step. In another example, each canonical instruction set may be a video of discrete steps to be performed by a human actor at a specific processing node or station 108.

[0080] At step 654, control module 106 may receive image data of the tooling (e.g., station 108) from monitoring platform 104. For example, acquisition system 206 may receive image data of the assembly process at a respective processing node. In some embodiments, acquisition system 206 may receive V number of images, where V may represent the number of cameras of monitoring platform 104 that may record the assembly procedure at a specific processing station 108. Accordingly, each image of the V number of images may capture a different perspective of the tooling during processing.

[0081] At step 656, control module 106 may extract a subset of images from the obtained image data. For example, following receipt of the image data, acquisition system 206 may be configured to extract a subset of images or frames. For example, acquisition system 206 may be configured to extract L number of images, i.e., landmark frames, from the received image data. The extracted images may be referred to as landmark frames. Landmark frames may be those image frames that are in high motion and may contain the most meaningful features that clearly captures the progress of the manufacturing process. The extracted images may include those images or frames that include certain landmarks, $\vec{I}$, of the component, where $\vec{I} = [i_0, i_1, \dots, i_{L-1}]$, and may represent the entire manufacturing process for the component.

[0082] At step 658, control module 106 may extract one or more keypoints of the tooling from the landmark frames. For example, extractor module 208 may extract keypoints from the L number of images. Extractor module 208 may identify or extract K number of keypoints for a given input, $I_l \in \mathbb{R}^{w \times h}$, where $l \in [0, L - 1]$. As output, extractor module 208 may generate

a single vector $\vec{P} = [\vec{p}_0, \vec{p}_1, \dots, \vec{p}_{L-1}]$. This vector may include landmark representations of K number of (x, y) pairs which may be represented by $\vec{p}_i = [(x_0, y_0), (x_1, y_1), \dots, (x_{K-1}, y_{K-1})]$, where $i \in [0, L - 1]$.

[0083] At step 660, control module 106 may predict a final quality metric for the component based on at least the identified keypoints. In some embodiments, prediction module 210 may implement a long short-term memory (LSTM) model to output the estimate of the final quality metric.

[0084] At step 662, control module 106 may classify the component based on the final quality metric. For example, classification module 215 may classify the component into one or more classes based on the predicted final quality metric, $\hat{V}_c$. To classify the component, tooling module 202 may receive a canonical or desired final quality metric, V_c , for the component. Tooling module 202 may be configured to compare the predicted final quality metric, $\hat{V}_c$, to the canonical or desired final quality metric, V_c , to generate a delta, Δ_V . Depending on the value of Δ_V , classification module 215 may sort the component into one or more classes.

[0085] In some embodiments, method 650 may further include operations 664-668.

[0086] At step 614, control module 106 may determine if the class assigned to the component is an acceptable class. If, at step 614, control module 106 determines that component has been assigned to an acceptable class, the manufacturing process may proceed to the next processing station or node (e.g., step 616), in accordance with the original instruction set. If, however, at step 614, control module 106 determines that component is assigned to an unacceptable class, then, at step 618 control module 106 may adjust the processing parameters of at least one of a processing station k or one or more processing stations $(k + 1)$. In some embodiments, control module 106 may interface with station controllers in at least one of current station 108 or subsequent stations 108 to adjust their respective control and/or station parameters. In some embodiments, control module 106 may provide human manufacturers with updated instructions to be performed in at least one of a processing station k or one or more processing stations $(k + 1)$ of a production line. These adjustments may aid in correcting the manufacturing process, such that the final quality metric may be within the range of acceptable quality metrics.

[0087] Figure 7 is a flow diagram illustrating a method 700 of correcting a multi-step manufacturing process, according to example embodiments. Method 700 may begin at step 702.

[0088] At step 702, an instruction set may be provided to manufacturing system 102. The instruction set may be representative of a set of instructions for a manufacturing process to be carried out by manufacturing system 102. In some embodiments, the instruction set may be provided to each station 108. For example, each canonical instruction set provided to each respective station 108 may define the processing parameters for a specific manufacturing step. In another example, each canonical instruction set may be a video of discrete steps to be performed by a human actor at a specific processing node or station 108.

[0089] At step 704, control module 106 may identify information corresponding to a component at a respective processing node. In some embodiments, simulation module 216 may receive tooling information from tooling module 202. Based on the keypoints generated by tooling module 202, simulation module 216 may be configured to generate a surface model representing a state of the component $\hat{a}_i$ at a specific process step i . In some embodiments, the surface model may be represented as $S(\vec{a}_i)$. In some embodiments, simulation module 216 may further be configured to generate or estimate a quality metric of the component $\vec{a}_i$. From the surface model, simulation module 216 may be configured to generate a points model, $P(\vec{a}_i)$, representing specific coordinates of the component, $\vec{a}_i$.

[0090] At step 706, control module 106 may determine whether an irreversible error has occurred. For example, SGD module 212 may receive points model, $P(\vec{a}_i)$, from simulation module 216. SGD module 212 may determine whether an irreversible error, k , has occurred by comparing the points model, $P(\vec{a}_i)$, at step i to a canonical points model, $P(\vec{a}^*)$ of a canonical component, $\vec{a}^*$. SGD module 212 may be configured to detect an irreversible error by taking a Hausdorff distance between the points model and the corresponding canonical points model. An irreversible error may be present when the Hausdorff distance between the current component $\vec{a}_i$ and the canonical component $\vec{a}^*$, at the respective processing station or node, exceeds some threshold tolerance.

[0091] If at step 706, control module 106 determines that an irreversible error has not occurred, then the manufacturing process may proceed to the next processing station or node (step 705), in accordance with the original instruction set. If, however, at step 706, control module 106 determines that an irreversible error has occurred, then method 700 proceed to step 708.

[0092] At step 708, control module 106 may generate an updated set of actions to correct the irreversible error. SGD module 212 may construct a set of updated actions $\{\{r_{k+1}, t_{k+1}\}, \dots, \{r_N, t_N\}\}$ given a set of actions up to a point of error an irreversible error k . In some embodiments, this set of updated actions may be referred to as x_{tail} . The sequences

of steps or actions that preceded and included the error step k may be referred to as x_{head} . Together, x_{tail} and x_{head} may define a component $\vec{a}_i$. Based on x_{head} , SGD module 212 may solve for x_{tail} using a stochastic gradient descent method. The sequences of steps or actions that preceded and included the error step k may be referred to as x_{head} . Together, x_{tail} and x_{head} may define a component $\vec{a}_i$. Based on x_{head} , SGD module 212 may solve for x_{tail} using a stochastic gradient descent method.

[0093] At step 710, control module 106 may generate a predicted final quality metric for the component, based on the set of actions generated by SGD module 212. For example, GRU model 214 may be configured to predict a final quality metric for the component, $\vec{a}_i$, based on $x_{tail} \oplus x_{head}$, where $\oplus$ may represent a vector concatenation operator.

[0094] At step 712, control module 106 may determine if the predicted final quality metric is within a threshold tolerance of the canonical final quality metric. For example, the final quality metric generated by GRU model 214 may be compared against a canonical final quality metric to determine if x_{tail} is proper. If, at step 712, control module 106 determines that the predicted quality metric is within the threshold tolerance, at step 714, control module 106 may adjust the processing parameters of at least one of processing station k or one or more processing stations $(k + 1)$. In some embodiments, control module 106 may interface with station controllers in a station 108(k) or subsequent stations 108($k + 1$) to adjust their respective control and/or station parameters. In some embodiments, control module 106 may provide human manufacturers with updated instructions to be performed in at least one of a processing station k or one or more processing stations $(k + 1)$ of a production line. These adjustments may aid in correcting the manufacturing process, such that the final quality metric may be within the range of acceptable quality metrics.

[0095] If, however, at step 712, control module 106 determines that the predicted quality metric is not within the threshold tolerance, GRU model 214 may prompt SGD module 212 to generate a new x_{tail} . For example, method 700 may revert to step 708 for SGD module 212 to construct a new set of updated actions.

[0096] Figure 8A illustrates a system bus computing system architecture 800, according to example embodiments. One or more components of system 800 may be in electrical communication with each other using a bus 805. System 800 may include a processor (e.g., one or more CPUs, GPUs or other types of processors) 810 and a system bus 805 that couples various system components including the system memory 815, such as read only memory (ROM) 820 and random access memory (RAM) 825, to processor 810. System 800 can include

a cache of high-speed memory connected directly with, in close proximity to, or integrated as part of processor 810. System 800 can copy data from memory 815 and/or storage device 830 to cache 812 for quick access by processor 810. In this way, cache 812 may provide a performance boost that avoids processor 810 delays while waiting for data. These and other modules can control or be configured to control processor 810 to perform various actions. Other system memory 815 may be available for use as well. Memory 815 may include multiple different types of memory with different performance characteristics. Processor 810 may be representative of a single processor or multiple processors. Processor 810 can include one or more of a general purpose processor or a hardware module or software module, such as service 1 832, service 2 834, and service 3 836 stored in storage device 830, configured to control processor 810, as well as a special-purpose processor where software instructions are incorporated into the actual processor design. Processor 810 may essentially be a completely self-contained computing system, containing multiple cores or processors, a bus, memory controller, cache, etc. A multi-core processor may be symmetric or asymmetric.

[0097] To enable user interaction with the computing device 800, an input device 845 which can be any number of input mechanisms, such as a microphone for speech, a touch-sensitive screen for gesture or graphical input, keyboard, mouse, motion input, speech and so forth. An output device 835 can also be one or more of a number of output mechanisms known to those of skill in the art. In some instances, multimodal systems can enable a user to provide multiple types of input to communicate with computing device 800. Communications interface 840 can generally govern and manage the user input and system output. There is no restriction on operating on any particular hardware arrangement and therefore the basic features here may easily be substituted for improved hardware or firmware arrangements as they are developed.

[0098] Storage device 830 may be a non-volatile memory and can be a hard disk or other types of computer readable media that can store data that are accessible by a computer, such as magnetic cassettes, flash memory cards, solid state memory devices, digital versatile disks, cartridges, random access memories (RAMs) 825, read only memory (ROM) 820, and hybrids thereof.

[0099] Storage device 830 can include services 832, 834, and 836 for controlling the processor 810. Other hardware or software modules are contemplated. Storage device 830 can be connected to system bus 805. In one aspect, a hardware module that performs a particular function can include the software component stored in a computer-readable medium in

connection with the necessary hardware components, such as processor 810, bus 805, display 835, and so forth, to carry out the function.

[0100] Figure 8B illustrates a computer system 850 having a chipset architecture, according to example embodiments. Computer system 850 may be an example of computer hardware, software, and firmware that can be used to implement the disclosed technology. System 850 can include one or more processors 855, representative of any number of physically and/or logically distinct resources capable of executing software, firmware, and hardware configured to perform identified computations. One or more processors 855 can communicate with a chipset 860 that can control input to and output from one or more processors 855. In this example, chipset 860 outputs information to output 865, such as a display, and can read and write information to storage device 870, which can include magnetic media, and solid state media, for example. Chipset 860 can also read data from and write data to RAM 875. A bridge 880 for interfacing with a variety of user interface components 885 can be provided for interfacing with chipset 860. Such user interface components 885 can include a keyboard, a microphone, touch detection and processing circuitry, a pointing device, such as a mouse, and so on. In general, inputs to system 850 can come from any of a variety of sources, machine generated and/or human generated.

[0101] Chipset 860 can also interface with one or more communication interfaces 890 that can have different physical interfaces. Such communication interfaces can include interfaces for wired and wireless local area networks, for broadband wireless networks, as well as personal area networks. Some applications of the methods for generating, displaying, and using the GUI disclosed herein can include receiving ordered datasets over the physical interface or be generated by the machine itself by one or more processors 855 analyzing data stored in storage 870 or 875. Further, the machine can receive inputs from a user through user interface components 885 and execute appropriate functions, such as browsing functions by interpreting these inputs using one or more processors 855.

[0102] It can be appreciated that example systems 800 and 850 can have more than one processor 810 or be part of a group or cluster of computing devices networked together to provide greater processing capability.

[0103] While the foregoing is directed to embodiments described herein, other and further embodiments may be devised without departing from the basic scope thereof. For example, aspects of the present disclosure may be implemented in hardware or software or a combination of hardware and software. One embodiment described herein may be implemented as a

program product for use with a computer system. The program(s) of the program product define functions of the embodiments (including the methods described herein) and can be contained on a variety of computer-readable storage media. Illustrative computer-readable storage media include, but are not limited to: (i) non-writable storage media (e.g., read-only memory (ROM) devices within a computer, such as CD-ROM disks readably by a CD-ROM drive, flash memory, ROM chips, or any type of solid-state non-volatile memory) on which information is permanently stored; and (ii) writable storage media (e.g., floppy disks within a diskette drive or hard-disk drive or any type of solid state random-access memory) on which alterable information is stored. Such computer-readable storage media, when carrying computer-readable instructions that direct the functions of the disclosed embodiments, are embodiments of the present disclosure.

[0104] It will be appreciated to those skilled in the art that the preceding examples are exemplary and not limiting. It is intended that all permutations, enhancements, equivalents, and improvements thereto are apparent to those skilled in the art upon a reading of the specification and a study of the drawings are included within the true spirit and scope of the present disclosure. It is therefore intended that the following appended claims include all such modifications, permutations, and equivalents as fall within the true spirit and scope of these teachings.

CLAIMS

1. A manufacturing system, comprising:
 - one or more stations, each station configured to perform at least one step in a multi-step manufacturing process for a component;
 - a monitoring platform configured to monitor progression of the component throughout the multi-step manufacturing process; and
 - a control module configured to dynamically adjust processing parameters of a step of the multi-step manufacturing process to achieve a desired final quality metric for the component, the control module configured to perform operations, comprising:
 - receiving image data of tooling of a first station of the one or more stations;
 - determining, by a first machine learning model, a final quality metric for the component, based on the image data;
 - determining that the final quality metric is outside an acceptable range of the desired final quality metric;
 - based on the determining, generating, by the first machine learning model, an updated instruction set to be performed by at least one of the first station or a downstream station to achieve the desired final quality metric;
 - predicting, by a second machine learning model, an updated final quality metric for the component based on the updated instruction set; and
 - based on the updated final quality metric, providing the updated instruction set to the first station and the downstream station.
2. The manufacturing system of claim 1, wherein the second machine learning model is trained with a training set comprising a synthetic set of data.
3. The manufacturing system of claim 1, wherein the final quality metric cannot be measured until processing of the component is complete.
4. The manufacturing system of claim 1, further comprising:
 - determining, based on positional information of the component, that an irreversible error is not present.

5. The manufacturing system of claim 1, further comprising:
identifying a set of keypoints from the image data, the set of keypoints corresponding to position information of the tooling during processing at the first station.
6. The manufacturing system of claim 5, wherein the set of keypoints are used by the first machine learning model to determine the final quality metric.
7. The manufacturing system of claim 5, further comprising:
comparing the set of keypoints corresponding to coordinates of the component to a canonical set of keypoints corresponding to a canonical component.
8. A method for dynamically adjusting processing parameters of a step of a multi-step manufacturing process to achieve a desired final quality metric for a component undergoing the multi-step manufacturing process, comprising:
receiving, by a computing system, image data of tooling of a first station of one or more stations involved in the multi-step manufacturing process;
determining, by a first machine learning model of the computing system, a final quality metric for the component, based on the image data;
determining, by a computing system, that the final quality metric is outside an acceptable range of the desired final quality metric;
based on the determining, generating, by the first machine learning model, an updated instruction set to be performed by at least one of the first station or a downstream station to achieve the desired final quality metric;
predicting, by a second machine learning model of the computing system, an updated final quality metric for the component based on the updated instruction set; and
based on the updated final quality metric, providing, by a computing system, the updated instruction set to the first station and the downstream station.
9. The method of claim 8, wherein the second machine learning model is trained with a training set comprising a synthetic set of data.
10. The method of claim 8, wherein the final quality metric cannot be measured until processing of the component is complete.

11. The method of claim 8, further comprising:
determining, based on positional information of the component, that an irreversible error is not present.
12. The method of claim 8, further comprising:
identifying a set of keypoints from the image data, the set of keypoints corresponding to position information of the tooling during processing at the first station.
13. The method of claim 12, wherein the set of keypoints are used by the first machine learning model to determine the final quality metric.
14. The method of claim 12, further comprising:
comparing the set of keypoints corresponding to coordinates of the component to a canonical set of keypoints corresponding to a canonical component.
15. A non-transitory computer readable medium comprising one or more sequences of instructions, which, when executed by a computing system, causes a processor to perform operations for dynamically adjusting processing parameters of a step of a multi-step manufacturing process to achieve a desired final quality metric for a component undergoing the multi-step manufacturing process, comprising:
receiving, by a computing system, image data of tooling of a first station of one or more stations involved in the multi-step manufacturing process;
determining, by a first machine learning model of the computing system, a final quality metric for the component, based on the image data;
determining, by a computing system, that the final quality metric is outside an acceptable range of the desired final quality metric;
based on the determining, generating, by the first machine learning model, an updated instruction set to be performed by at least one of the first station or a downstream station to achieve the desired final quality metric;
predicting, by a second machine learning model of the computing system, an updated final quality metric for the component based on the updated instruction set; and

based on the updated final quality metric, providing, by a computing system, the updated instruction set to the first station and the downstream station.

16. The non-transitory computer readable medium of claim 15, wherein the second machine learning model is trained with a training set comprising a synthetic set of data.

17. The non-transitory computer readable medium of claim 15, wherein the final quality metric cannot be measured until processing of the component is complete.

18. The non-transitory computer readable medium of claim 15, further comprising:
determining, based on positional information of the component, that an irreversible error is not present.

19. The non-transitory computer readable medium of claim 15, further comprising:
identifying a set of keypoints from the image data, the set of keypoints corresponding to position information of the tooling during processing at the first station.

20. The non-transitory computer readable medium of claim 19, wherein the set of keypoints are used by the first machine learning model to determine the final quality metric.