Jeffrey A. Abell
GM Technical Fellow
Mem. ASME
Global Research and Development,
General Motors,
Warren, MI 38092
e-mail: jeffrey.abell@gm.com

Debejyo Chakraborty
Global Research and Development,
General Motors,
Warren, MI 38092
e-mail: debejyo.chakraborty@gm.com

Carlos A. Escobar[1]
Global Research and Development,
General Motors,
Warren, MI 38092
e-mail: carlos.1.escobar@gm.com

Kee H. Im
Global Research and Development,
General Motors,
Warren, MI 38092
e-mail: kee.im@gm.com

Diana M. Wegner
Global Research and Development,
General Motors,
Warren, MI 38092
e-mail: diana.wegner@gm.com

Michael A. Wincek
Global Research and Development,
General Motors,
Warren, MI 38092
e-mail: mike.wincek@gm.com

Big Data-Driven Manufacturing—Process-Monitoring-for-Quality Philosophy

Discussion of big data (BD) has been about data, software, and methods with an emphasis on retail and personalization of services and products. Big data also has impacted engineering and manufacturing and has resulted in better and more efficient manufacturing operations, improved quality, and more personalized products. A less apparent effect is that big data have changed problem solving: the problems we choose to solve, the strategy we seek, and the tools we employ. This paper illustrates this point by showing how the big data style of thinking enabled the development of a new quality assurance philosophy called process monitoring for quality (PMQ). PMQ is a blend of process monitoring and quality control (QC) that is founded on big data and big model (BDBM), which are catalysts for the next step in the evolution of the quality movement. Process monitoring (PM) for quality was used to evaluate the performance of the ultrasonically welded battery tabs in the new Chevrolet Volt, an extended range electric vehicle.
[DOI: 10.1115/1.4036833]

Keywords: manufacturing, big data, big models, problem-solving strategy, process monitoring for quality, acsensorization, quality control

1 Introduction

The convergence of developments in data acquisition, computing, and analysis/modeling have created a new capability, or set of tools, which collectively is called *big data* (BD), an amorphous but somewhat convenient term. The first applications of a new capability typically involve performing known tasks faster and better to produce higher quality output, or it allows existing problems to be fixed and eliminated as seen in the early applications of data mining to fault detection and quality improvement [1]. A new capability does not necessarily come with a strategy to create something new, but it often implicitly suggests a direction to pursue, the exact path not clear. Only after the path has been created and the project completed can the strategy be articulated. This paper describes such a project and a point of view that helped to create a new manufacturing quality philosophy. The technology is ultrasonic welding of battery tabs (UWBT) [2] for the Chevrolet Volt, an extended range electric vehicle. At the time of the creation of the Volt, UWBT technology was highly reliable but incompletely understood. One of the quality characteristics of the weld is its strength that can be determined by a destructive pull test. All the welds in the vehicle must be good for the electric motor to function. At the time of the development of the Volt, GM was in the midst of bankruptcy proceedings. Any failure of the Volt could potentially negatively affect the viewpoints of the negotiators and the customers, not only about the future of GM but also about the future of electric vehicles. Any change in the scheduled launch of the vehicle could have had a similar negative impact. A "perfect" vehicle had to be launched on schedule. Though just one piece in a big project, a quality control procedure for UWBT was needed to help accomplish that goal. A data-driven, empirical approach motivated by new capabilities in data, computing, and analysis was used to compensate for the lack of theoretical knowledge of the new welding process. This paper will adumbrate a BD style of thinking that emerged from the UWBT project and illustrate it with some examples from the Volt UWBT project.

The remainder of the paper is organized as follows: Section 2 introduces the concept of big data—big model (BDBM) and explains how it is a catalyst in shaping the new era of manufacturing that gave rise to the process monitoring for quality (PMQ) philosophy. The central theme of the paper, PMQ, is motivated and discussed in Sec. 3, with the production of Chevrolet Volt battery pack as an example of successful implementation. Finally, Sec. 4 shows how PMQ may contribute to the total quality movement.

2 Big Data—Big Models

The BD environment has three basic interconnected components: data, computation, and analysis, as shown in Fig. 1.

[1]Corresponding author.

Manuscript received January 31, 2017; final manuscript received May 17, 2017; published online August 24, 2017. Assoc. Editor: Ivan Selesnick.

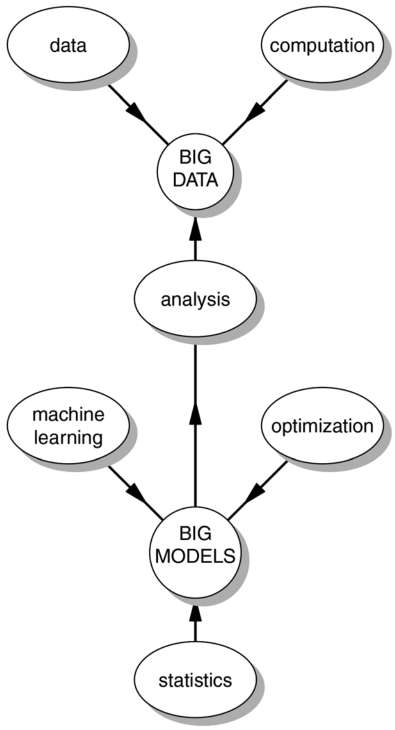


Fig. 1 Big data—big models

The three labels should be interpreted broadly so that they connote more than the label would otherwise indicate. Data include the technologies for data acquisition; hence, sensors and other measurement devices are in this category. Computation includes the technologies for storage, retrieval, transmission, and networking. Analysis includes knowledge discovery techniques as well as traditional statistical models and testing approaches. BD has been described in terms of three V's: *volume*, *variety*, and *velocity*, where the focus is on the data component. The term "data science," rather than "computer science," is used when the focus is on the compute component. The analysis component has no new name at this time, but it deserves one because new capabilities and possibilities have arisen from the synergy of machine learning, statistics, and optimization, as depicted in Fig. 1 where we call it *big models* (BM).

When the focus is shifted to BM, volume and variety still apply but with a slightly different meaning. The new modeling paradigm includes a discovery aspect that often requires many models to be created in order to find the final model. The challenge is to select the best single model or select the best group of models to create an ensemble—a set of models and a decision combination/fusion rule [3]. For example, in a random forest, multiple classification trees are created by bootstrapping the training data and sampling the features [4]. An item receives a classification based on the majority vote [5] of the constituent trees. This is a *homogeneous ensemble* because all the classifiers have the same base model, a classification tree. A *heterogeneous ensemble* is formed when the classifiers are not all the same. For example, the set of classifiers might include the support vector machine, logistic regression, and Fisher's linear discriminant. The homogeneous classifiers have volume; the heterogeneous classifiers have volume and variety.

These two V's for analysis lead to the need for model selection and/or model fusion. This, in turn, requires two more V characteristics as shown in Fig. 2: model *verification* (validation and testing) and *vigilance* (routine monitoring and updating the model). Verification is needed when few or no assumptions are made about the underlying process. In that case, one conceptually uses three data sets[2]

[2]Techniques such as cross validation address the situation where three distinct sets are not available.

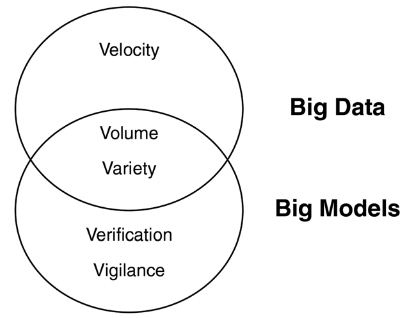


Fig. 2 The five V's of big data—big models

(training, validating, and testing) to arrive at a final model [6]. The modeling process requires that a proposed model be tried on new data and satisfy certain performance criteria before it can be deployed. Vigilance is required, especially in a manufacturing context, because the model is surely "wrong" but may be "good enough" for a while. Vigilance helps determine how long a "while" is and it implies that the model building never really ends.

Volume and variety can also have a different meaning for data, especially when the data comes in the form of signals or time series. Data can come from one sensor or from a variety of different sensors. Variety here comes from the different types of sensors and, in manufacturing, from the locations and times of their use. In other words, data gathering for products may start in the manufacturing plant, but it does not end there. It continues for the life of the item. Data gathered while the item is in use not only can be used to predict a possible failure or a need for maintenance, but it can also be used to better assess the quality of the item. This information can be used in the near term to make adjustments in the factory and in the long term by engineering and design to eliminate problems.

The process of deploying sensors everywhere is called *acsensorization*[3]. In the new environment where many sensors are available and affordable, it makes sense to use them for knowledge discovery. The potential gains can be significant, at a relatively low cost. Note that the acsensorization concept includes people as sensors. The smartphone technology with apps, cameras, voice recording, data transmission and the database technology with unstructured data make the person an even more powerful sensor. The choice of sensors is crucial to the success of the data analysis. Even though detailed theory may not be available for engineers to know exactly which features are related to the product performance, engineers must pick the sensors related to the fundamental mechanisms underlying the process so that reliable features can be extracted. Though the emphasis here is on data and models, as pointed out by George Box, the whole process "requires at each iteration, the injection of subject matter knowledge by the engineer or other specialist" [7].

In the Volt UWBT problem, data came in the form of signals from sensors, plant logs, etc. Signals may be used directly in a time series analysis, or features may be extracted from the signals and used as variables in models such as logistic regression. Features from a signal can include, for example, points, derivatives, integrals, durations, number of cycles, and times. Figure 3 gives a high-level view of the transformation of signals to features to classifiers to the final rule that gets deployed in the plant. The actions leading to the creation of the rule are typically done offline. The rule is computed on-line for each item to give a decision or a prediction, such as the class of the item produced.

[3]acsensorize [from accessorize] *v.t.* act of adding a multitude of dissimilar sensors, generally of a variety of sensing modalities, to an existing system that may or may not already have sensors; acsensorizing (*pres. part.*); acsensorized (*pass. part.*); acsensorization *n.* the process of acsensorizing.

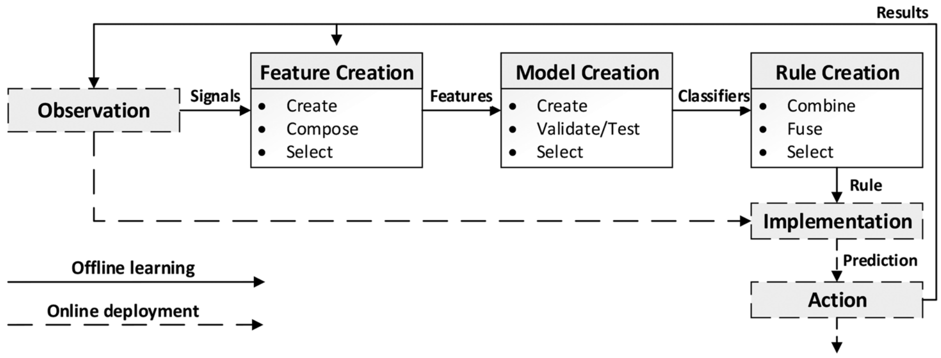


Fig. 3 Big modeling

Though Sec. 3 will explain and illustrate the boxes in Fig. 3, it is useful to emphasize the role of the verb *select* that appears in each of the three boxes related to the model building, because *selection* constitutes a theme that pervades the model building process. The verb in each box operates on a different noun: signals, features, and models. Selection can also occur with sensors, but that is not depicted in Fig. 3. From a multitude of sensors, possibly only a subset is used. Each sensor may produce a multitude of signals from which only a subset may be used. The signals typically produce very many features, some of which may be redundant, noisy, irrelevant, or even misleading. Only features that contribute to the predictive model should be selected. Finally, a large set of candidate models are created from which the final one(s) are selected for an ensemble. At each stage a big set is created and then distilled to a smaller set. We capture this expansion–contraction cycle in a mnemonic which we call the "bellows chart" that is given in Fig. 4. The cycle can occur in different forms as shown with respect to feature selection in Fig. 5. The multitude of models is a key characteristic of BM. As a result, selection is a key operation.

Big data—big models creates a new context for problem solving. The original context was front loaded: all the thinking, planning, and model selection had to be done before data were collected and analyzed. The goal was to build a set of facts as part of a theory of a larger body of knowledge. The focus was long term. In contrast, the BDBM context is back loaded: data are collected and analyzed first in order to create a model. The goal is pragmatic: to generate useful information that can be exploited, for example, to keep a process running or to ensure that the current output meets quality requirements. The focus is short term. The hope is that this information allows a stable body of knowledge to form, in which case the original context with the longer term focus can be employed. The BDBM environment is well suited to exploratory projects, but it can also be employed in problems without a clear starting point. It can provide suitable working solution until a permanent solution can be developed. One may prefer to go the traditional route, but sometimes there is no choice when deadlines have to be met. The BDBM way of thinking provides an alternative route. This route does have risk, because there is no guarantee that the correct data have been collected and that the analysis tools can successfully discover the relevant features and build an adequate model. This paper reports one successful instance.

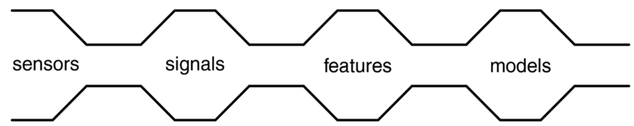


Fig. 4 The bellows chart: a mnemonic for selection in model building

3 Process Monitoring for Quality

3.1 The Process, the Product, and the Problem. Ultrasonic welding [8] is a joining process where entities that are in physical contact are joined by rapid relative motion between adjacent surfaces. An ultrasonic welder is a device that generates vibrations in a transducer assembly and applies them to the work pieces via a knurled sonotrode/horn. The work pieces are supported by a stationary knurled anvil. The sonotrode presses against the anvil, with the work pieces between, and vibrates (tangentially to the surface of the work pieces). Hereafter, the sonotrode and the anvil will be collectively referred to as the *tool*. Figure 6 shows these components in a schematic of an ultrasonic welding application for battery tabs [9–11]. The battery cells used in the Volt battery come as pouches with tabs for electrical connection as shown in Fig. 7(*a*). A cell group consists of three cells that are stacked together (electrically in parallel) and physically joined at the tabs to a busbar on an interconnect board as shown in Fig. 7(*b*). All the cells that were joined to the same interconnect board constituted a module, and the busbar provided the series current carrier for the cell group in a module. The final battery pack (Fig. 7(*c*)) was a series connection of such modules. Because the connections were in series, every single weld had to perform for the battery to function. Each weld had to meet two criteria: one was the mechanical strength of the weld and the other was its electrical conductance. The definitive test for strength is to pull the welds apart and note the force. The immature theory for UWBT did not immediately suggest a nondestructive proxy that could be used to infer strength. Similarly, an electrical test was available at the module level but not at the cell level[4]. The module level test could not identify the specific cells that caused the module to fail. An effective manual test for a weld was available, but it was laborious, inefficient, ergonomically challenging, and only suitable for very low volumes. The manual test was part of the system, but it could not be used for every weld. The UWBT process, even though highly reliable, would have been deemed immature by traditional standards and production would have been delayed.

The dilemma was how to construct immediately a quality assurance program for a highly reliable product whose performance characteristics could neither be directly observed nor indirectly inferred based on current understanding of UWBT. A solution required a different point of view and a different set of tools. BDBM set the stage for PMQ that provided both.

3.2 Process Monitoring for Quality (PMQ). Usually the development of a statistical product quality control (QC) procedure is somewhat routine. It is built on well-known theory and employs product quality characteristics that are directly and nondestructively observable to assesses the fitness of the product for the intended purpose. For example, a common quality

[4]A cell level electrical test was developed over the subsequent years that could infer electrical conductivity, but PMQ remains as the in situ total quality predictor.

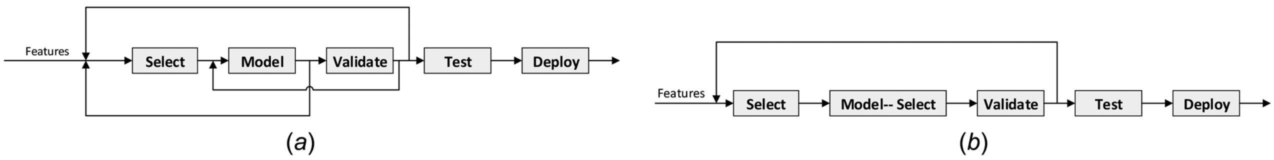


Fig. 5 Feature selection methods in classification: (*a*) filter and wrapper methods and (*b*) embedded methods

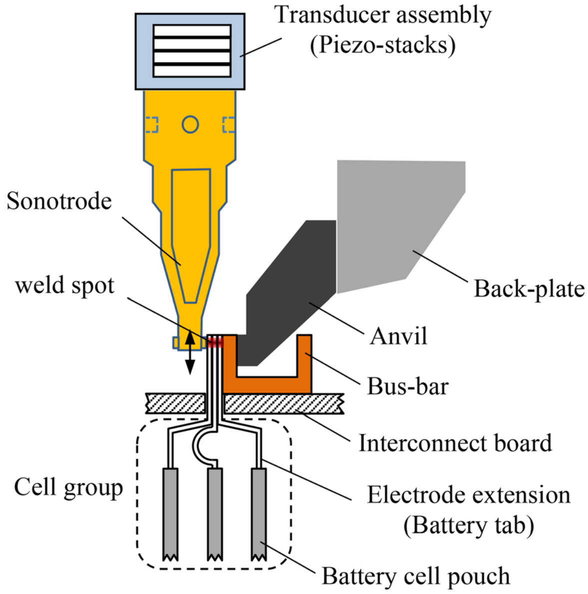


Fig. 6 Ultrasonic welding schematic for battery tabs (see Ref. [9])

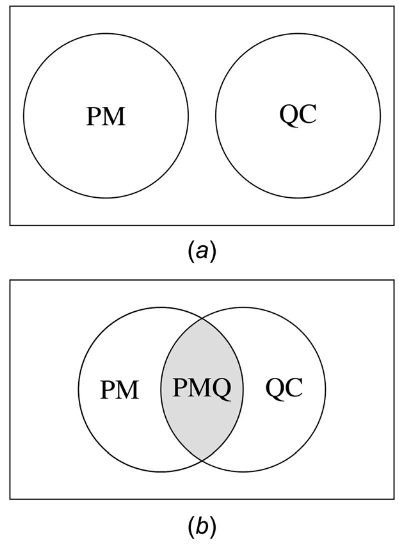


Fig. 8 Process monitoring for quality: a blend of process monitoring and quality control: (*a*) traditional view and (*b*) updated view

characteristic is a physical dimension, such as a length, which is usually directly observable in a nondestructive way. If the characteristic is not directly observable, then one relies on existing theory of the subject matter to create a proxy and obtain an indirect measurement. As mentioned, the dilemma with the battery tabs problem was that neither route was available. Hence, the goal was to fabricate a QC system that would work commendably, while more knowledge regarding the system could be gathered.

Since data from the product were not available, a logical recourse was to use data from the process. This is the domain of process monitoring (PM), which is related to but complementary to QC. PM data are typically specific features and indicators based on engineering knowledge and expectations of the process. Knowledge of the process is implicitly built into the data collected about the process. For a problem like the UWBT, the data had to be more fundamental because it would be the basis for learning. The fundamental data would be measurements on the process for each item as each item was being made. This implies a very large volume of data, which could be addressed successfully by the BDBM framework. For the Volt UWBT problem, this meant that appropriate sensors had to be chosen to observe the process. Section 3.3.1 discusses this. Gathering the large volumes of relevant process data was the first step, but this step would be fruitless if it were not possible to relate data from the process to quality characteristics about the product. Moreover, the learning rate had to be high as volume ramps up after product launch. This was enabled by the BM part of the BDBM environment.

Note that a key enabler to the project was the purposeful collection of fundamental data, new data. Existing plant data, though voluminous and varied, are collected for a different purpose and, hence, may not contain the needed information. Collecting the right data is dependent on the right sensors in strategic locations that can only be chosen by subject matter experts.

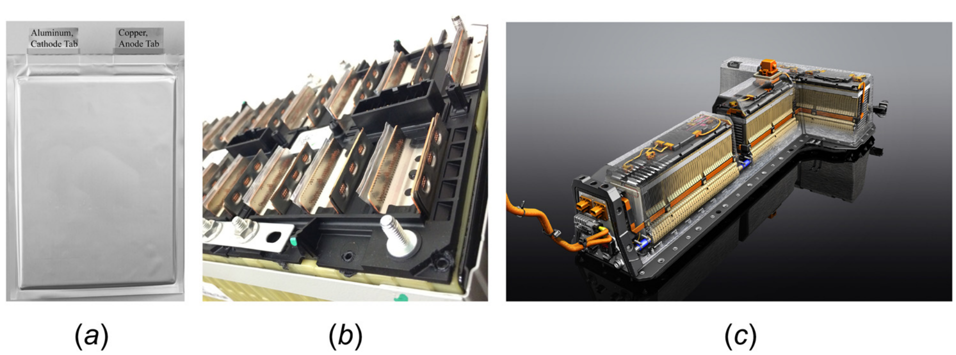


Fig. 7 The first generation Chevrolet Volt battery: (*a*) cell, (*b*) module, and (*c*) graphical rendering of the battery

The approach that was adopted to meet this challenge was a blend of PM and QC and called process monitoring for quality (PMQ) as qualitatively depicted in Fig. 8. The PMQ approach encompasses the PM philosophy of monitoring observable aspects of the process and the QC philosophy of predicting the fitness of the product. That it is not a true QC system is apparent in its output. Whereas a QC system seeks to declare each item as "good" or "bad," the PMQ system seeks to find the obviously good and declare them as good, and to declare all the others as "suspect." If a weld was declared good, it was thought to almost certainly be good and, hence, did not need to be inspected. If it was declared suspect, it might have been good or bad; only an inspection could ascertain. The suspect items were subjected to the manual test. As a result of the hedging, not coming to a definitive good/bad decision, the PMQ system in this application was effectively a filter for the manual test. Its goal was to keep the manual inspection rate at a level commensurate with its limited capacity.

This hedging is part of the price paid for the lack of direct measurements and for an incomplete knowledge base. The hedging also reflects a different perspective about the implications of making an incorrect decision. Recall that, at the time of the introduction of the Volt, it was vitally important that all vehicles perform as intended. From the quality perspective at the plant, this meant that the probability of declaring a bad weld good should be zero, even at the cost of declaring a good weld bad. The first error is called a *type II error* in statistics, a *miss* in signal processing, or a *false-negative* elsewhere. The second error is called a *type I error* in statistics or called a *false alarm* in signal processing, or *false-positive* elsewhere. These two types of errors are part of any detection system [12]. For a given detector, the two error rates, α for a type I error and β for a type II error, are inversely related. Since the two error rates cannot be simultaneously minimized in a given system, a deliberate trade-off must be made or, operationally, one error rate is specified, usually α, and the other is tolerated. The type I error rate, α, is usually set by the producer to limit the amount of resources expended on issues that are not really problems. The type II error focuses on the customer and this is also the focus of PMQ. With the philosophy and goals of PMQ established, Secs. 3.3.1–3.3.6 discuss the details by explicating the steps presented in Fig. 3.

3.3 A BDBM Implementation of PMQ for the Volt

3.3.1 Observation. The first step in addressing the PMQ problem was to assess existing process data to determine if it was sufficient to determine weld quality. A primary source of data comes from the welder controller. The operational mode of the controller is determined by user-specified parameters. The controller outputs a power signal to indicate the faithfulness of the performance to the specified mode. Analysis of the factory supplied low-resolution power curve did not yield a rule that could discriminate good welds from bad to the required accuracy. Basic engineering knowledge and experience produced the following facts:

- The power curve supplied to the user was a filtered subset of the power curve that was actually observed.
- A "cold" tool produces a different quality weld than a weld from a "warm" tool; warm is better than cold.
- During laboratory experimentation some users could actually hear a sound during the welding process and then use that sound to predict the state of the weld with an accuracy beyond random guessing.
- During the welding process, the sonotrode presses against the anvil and compresses the work pieces. Too little pressure results in no weld; too much pressure destroys the weld.

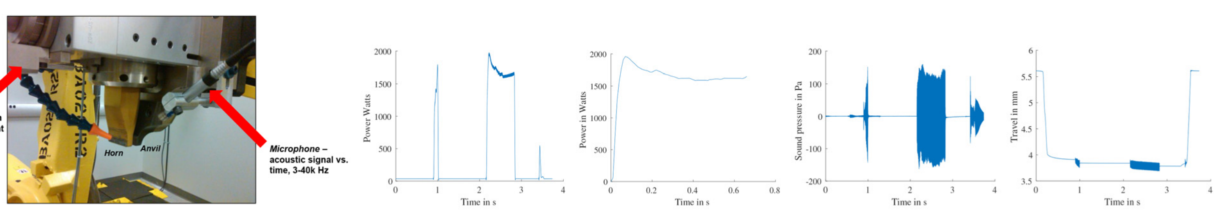
Acsensorization, based on the aforementioned facts, yielded the following sensors and example signals:

- The internal high-resolution power curve over the entire processing period was captured (Fig. 9(*b*)) in addition to the readily available low-resolution power curve of the welding (Fig. 9(*c*)).
- A temperature gauge was proposed (but not initially implemented).
- An acoustic sensor (a microphone) (Fig. 9(*a*)) was added to hear beyond the human range (Fig. 9(*d*)).
- A linear variable differential transformer (LVDT) (Fig. 9(*a*)) was added to measure in real time the compression (Fig. 9(*e*)) of the battery tabs.

The intent of acsensorization is to ensure that all the physical aspects of the process are captured so that information from the data could provide an insight into the process. In addition to the choice of sensors, the characteristics of the sensors, such as their range of sensitivity, and their placement and installation require engineering expertise. For example, an ill-placed acoustic sensor could easily measure background noise and provide no information. The signals from the sensors and the available plant data form the raw material from which models are created. Though the signals could be used in a time series analysis, most often features, or variables, are extracted to form a model. This process is the subject of Sec. 3.3.2.

3.3.2 Feature Creation. In a typical problem-solving situation, the collection of features to be used to create a model is given and finite, and available theory suggests the form of the model. The task then is to select the relevant features as inputs to the model. When the inputs are signals, such as the signals in Fig. 9, the features are not immediately available but must be computed from the signals. This is just an added step when available theory indicates what to compute, but a formidable step when the features are not known. In the latter case, the first task is to discover from the signals the features that relate to the outcome.

There are three sources of inspiration from which features can be generated: subject matter theory, descriptive statistical signal processing (SSP), and evolutionary methods such as genetic programming (GP) (see Fig. 10). The theoretical approach relies on existing knowledge to either create features that are known to be relevant, to suggest features based on the fundamental principles underlying the process, or to follow typical feature construction techniques [13]. The descriptive approach translates behavioral characteristics of the signals into quantitative variables. In the



(*a*) (*b*) (*c*) (*d*) (*e*)

Fig. 9 Acsensorization of ultrasonic welder and an example of observed signals: (*a*) some additional sensors to ultrasonic welder, (*b*) signal from power sensor (high resolution), (*c*) signal from power sensor (low resolution), (*d*) signal from microphone, and (*e*) signal from LVDT

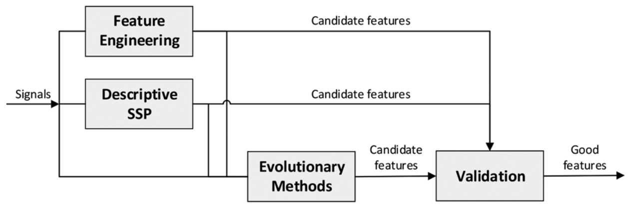


Fig. 10 Discovering features for big models

UWBT problem, the descriptive method created features such as the initial slope, the height of the first peak, and the duration from first peak to second peak from the low-resolution power signal, and the SSP methods created frequency related features from the acoustic signals.

The success of both of these methods depends upon prior knowledge and people's creativity. Selection of the final features is usually left to the model building process. Note that, for the most part, feature creation is a labor-intensive, inefficient, manual process. When product, process, or material changes are implemented in the production system, the analogous changes need to be reflected in the models, so rework is needed to search for and validate predictive models.

An evolutionary computing approach, such as genetic programming (GP), to feature creation in classification randomly creates a feature and then evaluates its predictive capability by means of a fitness function. Features can be composed from existing features, or they can be extracted from the original signal. An example of the former is the ratio of two features, and an example of the latter is the integral of a certain portion of the signal. As in evolution, the stronger features will tend to persist and improve through the generations or epochs, while the weaker features will tend to die out. The performance of the method depends on the catalog of composition and extraction methods and on the fitness function.

The composition and extraction methods include the common arithmetic functions (addition, multiplication, …), common operations in calculus (derivation, integration, …), and subject area-specific functions to customize the method to the problem. The evolutionary framework uses these operations and functions to randomly propose a new feature. The value or usefulness of the proposed feature is measured or quantified by the fitness function. The fitness function should be in alignment with the goals of the created features. For the UWBT problem, a fitness function, called the maximum probability of correct decision (MPCD) and described in Sec. 3.3.3, was created to evaluate the performance of a feature in a binary classification situation where one class occurs infrequently. A common measure, such as prediction accuracy, is not adequate because it would always predict the dominant class. The Pareto multi-objective GP produces a collection of features which are characterized by their prediction performance

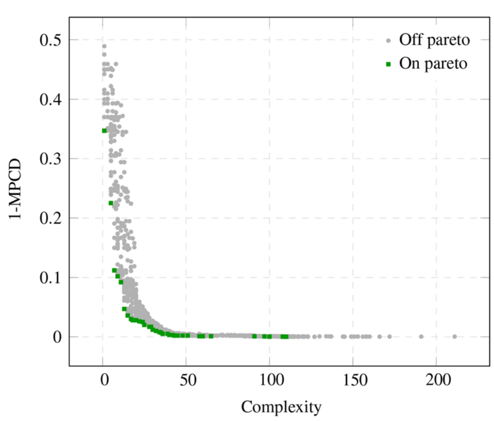


Fig. 11 Multi-objective pareto optimization

and their complexity as shown in Fig. 11, where smaller is better for both measures. Features with points on the knee of the Pareto front are candidates for inclusion in the model.

Whereas this process of feature proposal and feature evaluation was previously only done by humans, the evolutionary framework allows this to be done by the computer. Where the human can only afford to try a few promising ideas, the computer can try many more possibilities and, hence, sift through a large "quantity" of proposals to get a quality feature. Features that evolve to an acceptable level of fitness are then added to the set of candidate features for model building, which is the next step. This method is only practical now because of the computational power of the BDBM environment.

3.3.3 Model Creation. Practical modeling has always been a mixture of both the theoretical and the empirical. Prior to BDBM, the theoretical outweighed the empirical. This is the standard approach. Theory is used to generate a model that is specified before data are collected. The purpose of the analysis is to estimate the model parameters and, as a result, to accept or reject the model. Since the model building is done before the collection of data, the standard approach is front loaded. BDBM enables a back-loaded approach: collect data first in order to create a model. Theory for the standard approach is well developed and mature. Theory for the new approach is under development.

The practice of using data to create a model under the standard approach was given the name "data snooping."

> Data snooping occurs when a given set of data is used more than once for purposes of inference or model selection. When such data reuse occurs, there is always the possibility that any satisfactory results obtained may simply be due to chance rather than to any merit inherent in the method yielding the results. It is widely acknowledged by empirical researchers that data snooping is a dangerous practice to be avoided, but, in fact, it is endemic [14].

A more neutral term is "specification search," a "data-dependent process of selecting a statistical model" [15]. There is also another name.

> Data snooping is also known as data mining. Although data mining has recently acquired the positive connotations as a means of extracting valuable relationships from masses of data, the negative connotations arising from the ease with which naive practitioners may mistake the spurious from the substantive are more familiar to econometricians and statisticians [14].

The problems, and disappointments, arise when classical statistical theory is applied to models derived from a data mining process. As the word "endemic" in a previous quote indicates, BDBM is not the instigator of the iterative use of the data to specify a model. It was widely done "in private"; now with data mining, it is done "in public." Whereas one knows, for example, how to calculate confidence values, levels of significance, and the values of type I and type II errors under the standard scenario, those same calculations are not correct under the data mining scenario, because the data mining scenario violates the assumptions of the standard scenario.

In standard statistical theory, model building is a one-pass-through-the-data process. Multiple passes allow for different methods. Three main approaches have been developed for feature selection for classifiers: the filter method, the wrapper method, and the embedded method.[5] In Fig. 5(*a*), the strategy with no feedback loop to the select operation is the filter method. The others are variations on the wrapper method. Figure 5(*b*) shows two possible embedded methods.

Filter methods select features independently of the model fitting algorithm or its error criteria. Wrapper methods use the results of a fitted model to evaluate the performance of the features in the

[5]In this scenario, the model form is fixed and the features must be selected.

model and iteratively find a feature subset that best meets the performance criteria. The model fitting algorithm and the selection algorithm are separate but coupled in a feedback loop. In an embedded method, the model fitting and feature selection occur together. An example of this is logistic regression with ℓ_1 regularization [16]. The drawback of these iterative techniques is that they may learn the training data set very well but fail to generalize to a new data set and, thus, have poor performance where it matters the most.

The new strategies do not come with conditions under which confidence statements can be made about their performance. Each strategy generates a number of fitted models from a training set. A validation data set is used to select from the aforementioned sets one model for potential deployment. The performance of the selected model is evaluated using a test data set. If the model meets the performance criteria and if nothing has changed in the modeled process, the model is deployed. This procedure is different from the classical theory for which the validation comes from the theory of mathematical statistics. In the classical context, the model results are valid if the assumptions are valid. In this *ideal* (but unrealistic) context, no test is needed. In reality, a test would serve as a check on the assumptions.

A new binary classifier was developed for the UWBT problem. The UWBT process had a very high conformance rate, so a bad weld was a rare event. A key assumption to the development of the classifier is that the bad items are manifested as outliers with respect to key, but unknown, features. This is a reasonable assumption in a manufacturing situation where efforts are constantly made to keep the process stable. It is also consistent with the control chart point of view where an observation outside the control limits is a possible indicator of a problem. Control charts use probability theory to determine the limits. In the problem of interest, the extreme imbalance of data (too many good, too few bad) limits the use of probability. Instead the approach here was to use geometry.

A binary classifier is a partition of a set into subsets, where each subset is associated with one of the two classes. A point receives the label, or class, of the subset to which it belongs. Here, the labels are good and suspect. Hence, an item whose data features are in the good region is classified as good. Analysis of the UWBT data failed to yield one feature that could adequately and consistently separate the good welds from the bad. Multiple features had to be used. Just as a linear classifier partitions a space into two regions by means of a hyperplane, the initial goal for the UWBT classifier was to partition the space into two regions, the good region and the suspect region, where the good region has the following ideal characteristics: (a) *only* good are inside the good region; (b) *all* the bad are outside the good region; and (c) *some* good may be outside the good region. With respect to the data in the training set, the second characteristic says that all bad items (in the training set) are declared bad so that a bad item is never called good. This is equivalent to $\beta = 0$, a zero false-negative rate. The other characteristics say that it is possible for a good item to be declared bad, or, equivalently, $\alpha \geq 0$, a nonzero false-positive rate.

The aforementioned characteristics do not dictate the shape or structure of the good region. Since the only assumption about the bad items is that they are "outliers" in some sense, there is no justification to creating a complicated good region. A simple region in multiple dimensions based on a direct generalization of the form of a univariate control chart, an interval, is the Cartesian product of the intervals. The Cartesian product of two intervals is a rectangle. The Cartesian product of three rectangles is a box. The Cartesian product of any finite number of intervals, say m, is closed polyhedron whose sides are parallel to the axes. This can be envisioned as a "box," $\mathbb{B}$, in m dimensions, where $\mathbb{B} = \mathcal{I}_1 \times \mathcal{I}_2 \times \cdots \times \mathcal{I}_m$, with $\mathcal{I}_h = [\ell_h, u_h], h = 1, \ldots, m$.

The method for finding the box is a sequential use of a multiple comparisons procedure [17]. Each step involves finding an interval for each feature along with a measure of the goodness of the interval with respect to the goal of separating the two classes. We call the measure the maximum probability of correct decision (MPCD), to be defined shortly. The interval of the feature with the largest MPCD is used in the classifier. All items whose value of the chosen feature is outside the interval are excluded from the data set. This concludes the step. If the resulting data set contains any bad items, the aforementioned process is continued and another step is taken. The process ends when the resulting data set contains no bad. The Cartesian product of the intervals forms the box so that the classification rule is

$$B(\mathbf{x}) = \begin{cases} 0 \ (\text{"good"}) & \text{if } \mathbf{x} \in \mathbb{B}, \\ 1 \ (\text{"suspect"}) & \text{if } \mathbf{x} \notin \mathbb{B}. \end{cases}$$

The MPCD, the performance measure of the goodness of a feature, is a direct application of the idea of hypothesis testing in a simple case. Suppose the values of the feature are x_i, $i = 1, \ldots, N$, with class membership given by

$$c_i = \begin{cases} 0 & \text{if item } i \text{ is good}, \\ 1 & \text{if item } i \text{ is bad}. \end{cases}$$

Let n_g be the number of good items in the current data set and n_b be the number of bad. For each distinct value, say v, of the feature, let $\{x | x \leq v\}$ be the "call good" region, and its complement be called the "call suspect" region, and calculate

$$L(v) = \max_v \left(\frac{1}{n_b n_g} \sum_{i=1}^{N} I(x_i \leq v, c_i = 0) \sum_{i=1}^{N} I(x_i > v, c_i = 1) \right)$$

where $I(s)$ is the indicator function which returns 1 when s is true and 0 when s is false. Similarly, reverse the roles of the two open intervals so that $\{x | x \leq v\}$ is the "call suspect" region, and its complement is the "call good" region, and calculate

$$U(v) = \max_v \left(\frac{1}{n_b n_g} \sum_{i=1}^{N} I(x_i \leq v, c_i = 1) \sum_{i=1}^{N} I(x_i > v, c_i = 0) \right)$$

Let $v_u = \arg\max_v L(v)$, $v_\ell = \arg\max_v U(v)$, $v_{\min} = \min_v v$, and $v_{\max} = \max_v v$. If $L(v_u) > U(v_\ell)$, then the interval is $[v_{\min}, v_u]$ with $\text{MPCD} = L(v_u)$, otherwise the interval is $[v_\ell, v_{\max}]$ with $\text{MPCD} = U(v_\ell)$.

The construction of the box is a special form of a classification tree where the "splitting function" is the procedure that finds the point associated with the MPCD [18] and where the focus is on growing one branch that terminates in a pure node containing only good items. Since this classifier is a classification tree, it is deployed as such. In particular, it can be deployed as a random forest.

The UWBT process is inherently a changing process because, for example, the knurls on the anvil and horn are constantly being worn. Hence, the classifier must be changed as the process changes. The trigger for relearning a classifier is the "suspect rate." A rise in the suspect rate indicates that the process has changed. Since suspect items are inspected, the plant can learn if the process has changed due to normal conditions or whether an actual problem has emerged. Vigilance is required to watch the suspect rate and then relearn the process as needed. The method provided information about both the process and the product.

3.3.4 Rule Creation. An individual classifier, containing a subset of features, methods, and decision guidelines or thresholds, has shortcomings. It is often not able to provide the required decision error performance, even if it is good at detecting certain characteristics of the item. Attempting to tune a classifier to indefinitely reduce error can also lead to over-fitting. In addition, a classifier works best on certain features. So, one can be quickly posed with the problem of selecting classifiers and the best features to use with them. Such challenges have a natural solution.

Rules could be created by making a judicious ensemble of classifiers to provide the final decision.

Ensemble methods [5,19,20] and multiple classifier systems (MCS) [3,21] are powerful solutions to complex classification problems. The classification performance of a single classifier can be improved by combining many of them. The basic idea is to develop a set of either homogeneous or heterogeneous classifiers, evaluate their selection criteria, and combine the qualifying ones. The evaluation criteria has various options [22,23] which could include a very rudimentary criteria like $\max(\alpha, \beta) < 0.5$, to something more procedural like a dependency analysis [24]. The need for a valid selection method is extremely essential, because a sufficient number of poorly performing classifiers can adversely affect the ensemble classification performance. An optimal method to combine several independent classifiers is to use a Bayesian framework [25]. In that framework, the combined classification would outperform the classification of the best performing individual classifier. In reality, however, the independence is at best a fair assumption and the classifier suboptimal.

3.3.5 Implementation. PMQ software architecture development was posed with four primary challenges. First, due to the urgent nature of the Volt project, the time from research prototype to manufacturing plant implementation was in days. Traditionally, a research concept is developed, then a prototype is build, then a production system is created which undergoes rigorous testing, and only then it is deployed at a manufacturing facility. In our case, we needed a solution that could be rapidly deployed. Second, the software needed to be performed several simultaneous tasks. It had to perform complex mathematical computations in real time on large time series signals, handle communication with weld controller, and respond to the human user either by accepting an input or by displaying the acquired signals and status. This required a multithreaded implementation. Third, implementing the ever-evolving features, classifiers, and rules is a challenge, at least in the traditional approach of software architecture [26]. The underlying limitation of traditional software development is in the assumption that the mathematical formulation does not change. If the mathematics change, especially to previously unforeseen formulation, the software requires recoding, recompiling, and testing. Since the evolving features and classifiers are "changing mathematics," and they need updating as frequently as the offline routines discover feature(s) and classifier(s), we needed an architecture that allowed swapping mathematical calculations without rebuilding the entire software. Finally, the rules would also be updated continually. To keep the software up to date, we needed a technique to communicate that with the software without having to recompile any of its components.

To support the rapid progression from research phase to implementation, we adopted a multiplatform approach where different software development platforms were chosen to execute the task that they are best designed for. In our case, we used LabVIEW™ to handle the data collection and human–machine interface (HMI). The inherent multithreadedness of LabVIEW™ was specially suitable for designing a responsive HMI that could concurrently collect and manage large volumes of data and communicate with the weld controller. The HMI developed for the use in the UWBT application is shown in Fig. 12. We adopted a stratified architecture for our software, as depicted in Fig. 13, that naturally accommodated the *rapid-swap* modular programming philosophy that was needed to rapidly implement new features, classifiers, and rules. The features and classifiers were developed in MATLAB. The support of several well-designed toolboxes and the scripted programming approach in MATLAB made coding complex algorithms much easier than it would have been using the graphical programming approach that LabVIEW™ offers. In addition, MATLAB could deploy the codes in the form of dynamic-link library (DLL) on the Windows® operating system. A user could choose any operating system that supports runtime linking and loading of precompiled libraries (e.g., shared object libraries in Linux and

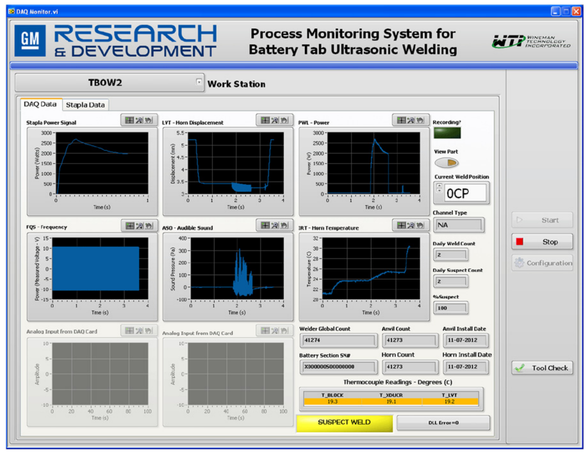


Fig. 12 Software interface to PMQ at Brownstown battery assembly plant

macOS™) and any programming environment that supports building such libraries.

The selected classifiers were compiled into a set of DLLs known as the *analysis library*. Any update to the classifier or a discovery of a new feature involves updating the analysis library files without compiling the whole software. A subsequent restart of the PMQ software would implement the new libraries. The rules are implemented through *configuration* files. The configuration files are ASCII encoded files that are read in at run-time, either at start or by polling for a change. An update to the rules entailed updating these files. To enable cross language and/or runtime interoperability between LabVIEW™ *graphical user interface* and MATLAB analysis library, we used *library wrapper* codes that were developed in C++, the strong suit of which is direct memory manipulation.

There are potentially two types of failures in the acsensorized PMQ system: (a) sensor failure and (b) setup failure. Either of these failures can increase the type I error, $\alpha \approx 1$, the latter being a genuine quality concern. The obvious solution to remedy the sensor failure was to incorporate a self-diagnostic routine that was aware of how the sensors are supposed to behave and exploit that knowledge to diagnose the sensors in real time. Setup failures are not something that is easy to detect in real time. To address such failures, the welding system was equipped with an off-line diagnostic capability [28,29].

3.3.6 Action. The process of quality inference ends with a prediction, but, without an action to follow, the prediction would serve no purpose. As suggested earlier, the good items are readily accepted and move to the next step in the manufacturing process, while the suspect items undergo further investigation and repair, if possible. The action steps were not different for UWBT.

After the PMQ software makes a prediction, the information is forwarded to an inspection station. However, among the several welds a module could have, it was impossible for a user to identify each suspect weld with precision when the information was communicated on a computer screen. The human inspector was required to inspect and report the status of each suspect weld as good or bad. This created room for human error, especially in a high-volume production environment. Once in a while the inspector could not correlate the correct physical weld location with the one identified on the screen. The accuracy of the inspection result was not only of paramount importance for a product quality assurance but also for training the PMQ system to improve the performance at the next epoch (Fig. 13).

To overcome this challenge, we introduced a visualization tool[6], an industrial overhead light projection system as described in Ref. [30] and depicted in Fig. 14. In the figure, a battery module

[6]Another 'V' in Big Data, but here the visualization uses hardware.

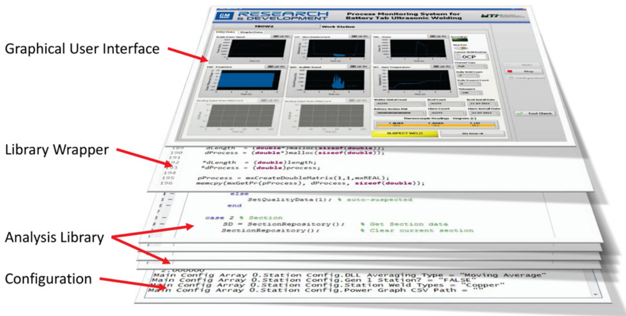


Fig. 13 Stratified software architecture used in PMQ

Fig. 14 Manual inspection station at Brownstown battery assembly plant (see color figure online)

with tabs illuminated with yellow and blue lights are shown. The suspect tabs are illuminated with yellow, and the inspector is supposed to check those. A corresponding diagram appears in a touch screen HMI next to the inspector (not shown in the figure) which could be used to provide the inspection outcome of good or bad. This greatly improved the inspection fidelity.

3.4 PMQ Summary. The original goal of the UWBT project was to reduce manual inspection of the battery tabs from 100% to 50%. The explicit purpose was not to improve quality but to accomplish two tasks:

(1) learn the weld process characteristics present at the time of weld creation that are associated with a good weld, and
(2) check that those characteristics are present during the creation of future welds.

The implementation took the form of a classifier where the two decision classes are good and suspect in contrast to the two quality

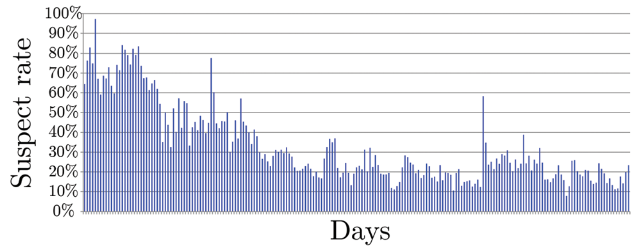


Fig. 16 Suspect rate over the span of the first year of implementation

classes of good and bad. A weld with characteristics outside the previously identified good region, a suspect weld, did not imply that the weld was bad, but that it could not be classified as good. Due diligence required that that weld be further investigated.

The classification task was challenged by the lack of theory and understanding of UWBT. PMQ provided the feasible alternative. The investigation philosophy of PMQ could be broadly divided into four phases of problem-solving strategy (a) acsensorize, (b) discover, (c) learn, and (d) predict, as depicted in the iconic representation of PMQ in Fig. 15.

Initial implementation of PMQ for UWBT used data from the power sensor. The low-resolution signal, which was provided by the welder manufacturer and shown in Fig. 9(*b*), captured data only during the actual welding, while the high-resolution signal, which was added by acsensorization and shown in Fig. 9(*c*), captured the preweld, weld, and postweld activity. The type and number of prebursts and postbursts were informative covariates. With these data and a rudimentary form of the box classifier in Sec. 3.3.3, the suspect rate went below the 50% target on production day 60, as shown in Fig. 16, and remained there with one exception that was due to a tooling problem. Further improvements in the suspect rate were achieved as the box classifier matured and as the other sensors with their associated features were employed.

A major challenge to the data-driven approach of PMQ is the choice of data sets for training, validation, and testing, especially when the environment is changing as in an initial launch when plant procedures have not converged to a standard practice and those variations are indirectly transmitted to the features. The data sets should be close in time to the period of application of the resulting classifier. One would like to use all the historical data, but, in a changing environment, the relevance of the data for building a model diminishes with time. The choice of times for each data set is problem specific. The training set must be relevant if the classifier is to be effective. An extreme problem occurs when the process has changed, relearning is necessary, and the historical data are not relevant. In that case, the data-driven approach dictates 100% inspection until a model is learned. This approach may not be feasible for the manufacturing plant, in addition to being very unpopular. The need to train, validate, and test

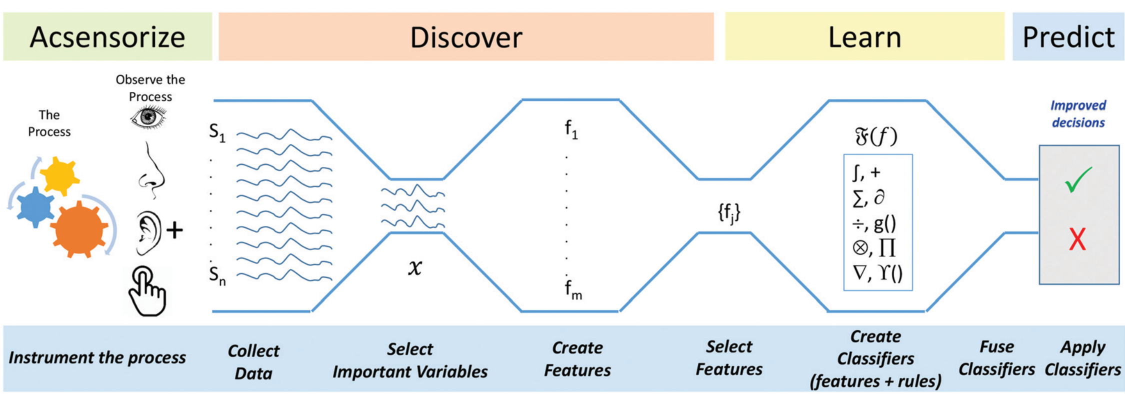


Fig. 15 Iconic representation of the PMQ philosophy

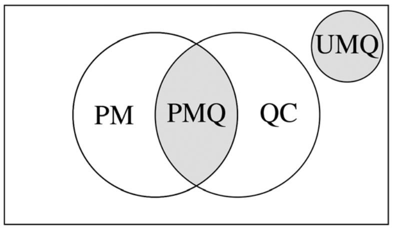


Fig. 17 Process monitoring for quality (PMQ): extended view

a model in as short a time as possible is an added complication in the manufacturing environment.

Another complication to empirical model building in manufacturing is that the model building process is an on-going activity. This is part of the price that must be paid for the lack of an engineering knowledge and for a changing plant environment. After the creation of the initial model, further model building is triggered by a rising suspect rate. Different responses are required depending on the result of the inspection and the locations of the features. For example, if the suspect items have been found to be good and the values of the features are not in the region of the training data, then one hypothesis is that plant practices have changed but this change has no effect on quality. A new classifier should be built to reflect the new situation, and this will lower the suspect rate.

While PMQ has been shown to address these concerns, it extends beyond this; especially in four broad scenarios. Implementation of the manufacturing processes assumes some risk of failure. This risk is significantly elevated in advanced and complex manufacturing processes that are not well understood, especially where there is a lack of knowledge on what characteristic(s) of the items to observe. In such processes a good item could only be identified by endorsing its functional success. PMQ could provide abstract, yet precise, mathematical indicators that can not only predict item quality but also aid in inferring salient aspects of the complex process. This is the first scenario. The second scenario is when the process is understood and we know the characteristics to observe, but there are no nondestructive techniques (NDT) available to observe such characteristics. The third scenario is encountered when NDT exists for the characteristics, but is not implementable in a manufacturing process because the technique is either slow or expensive. This should not imply that if we have implementable NDT methods to observe the necessary characteristics (aka QC), PMQ does not have anything to contribute. In the last scenario, PMQ is invaluable in a manufacturing system with deep buffer. In case of a quality spill,[7] the items in the buffer between the manufacturing system and the end of line quality inspection system would have to be discarded. In this scenario, PMQ empowers the manufacturing process with an in-place quality prediction tool, preventing such a spill and reducing scrap. These scenarios fall inside the intersection of PM and QC in Fig. 17.

PMQ can also be used to predict a failure of an item that is in use. The Internet of Things and acsensorization allow collecting and monitoring usage data that can help identify cohorts of items that share similar characteristics observed during manufacturing. If members of a cohort are observed to fail prematurely, the manufacturer can take one of the following steps to remedy the problem:

(1) the suspects can be intercepted in the field and called in for maintenance to prevent future warranty events, and
(2) the features can be analyzed to indicate changes in the process or product that may eliminate the problem altogether.

[7] A large number of successive items are produced with unacceptable quality.

Hence, the PMQ concept can be extended to usage monitoring for quality as shown in Fig. 17, a domain that is beyond PM and QC.

4 BDBM and the Next Step in the Quality Movement

This paper has referred to BD more as a way of thinking than merely as a collection of tools and capabilities. The BD way of thinking enables new strategies of learning, of problem solving, and of problem selection in manufacturing. The Volt UWBT story illustrates this at the project level. The BD style of thinking can, and should, affect the enterprise level. A natural conduit for these ideas to affect manufacturing is through the quality movement which has a presence in every manufacturing organization. Since its inception, the quality movement has continuously absorbed new ideas, techniques, and philosophies.

Modern quality control began in the 1930s by Dr. Walter Shewhart who merged statistics, engineering, and economics to develop a new industrial statistical quality control (SQC) theory (Fig. 18(*a*)) [31]. He also articulated a problem-solving strategy based on the scientific method that could be shared by both the worker, the problem solver at the microscopic level, and management, the problem solver at the macroscopic level. This strategy is known as the Shewhart learning and improvement cycle [32], Table 1-(a). His approach set a style that his successors followed.

Total quality management (TQM) emerged in the 1980s. As the name implies, the emphasis was on management. The management focus was not only on the product but also on the process and the people, both customers and employees (Fig. 18(*b*)) [33]. Shewhart's problem-solving strategy was enhanced and refined by Deming (Deming cycle), Table 1-(b), and seven basic quality tools were developed to make problem solving more accessible and systematic. Note that this new development embraced and extended SQC. This is another style that others would follow.

Six sigma was introduced by Motorola engineer Bill Smith in 1986. The Deming cycle was adapted again, but the main emphasis was on eliminating defects in all processes—manufacturing, service, transactional—by identifying and removing sources of variation. Six sigma greatly impacted the quality movement at the deployment level by incorporating project management techniques, using high-performance teams, and tracing project-results to the bottom line [34,35]. Where six sigma is mainly a reactive approach that seeks to identify and remove causes of defects and/or sources of variation, its complement is the design for six sigma (DFSS), Fig. 18(*d*), a proactive approach that seeks to design quality into products and processes so that defects never arise, Table 1-(d) [36,37].

The BDBM environment is a catalyst for the next step in the evolution of the quality movement, Fig. 19. The question is what will be its form in the 21st century. Management is certainly aware of BDBM as is evinced by the development of positions such as chief data officer and chief analytics officer in addition to chief information officer. However, the need for and the exact responsibilities of these positions is under active debate. At the worker level, a different issue arises. The methodology of the quality movement does not preclude the use of advanced methods, but it has always focused on common sense methods that could efficiently be used by the workers who are closest to the actual problems. Proper training and deployment are critical elements to the success of the quality movement. George Box [7] referred to the quality movement as the "democratization and comprehensive diffusion of simple scientific method." However, now the situation is not "simple." The new analytical techniques and the supporting data science may threaten that democratization since the new methods are more theoretically and computationally complicated. What do problem solvers really need to know to use the new tools? Part of the answer may come with the automatization of parts of the model building process. The DARPA data-driven discovery of models (D3M) [38] program "aims to develop automated model discovery systems that enable users with subject

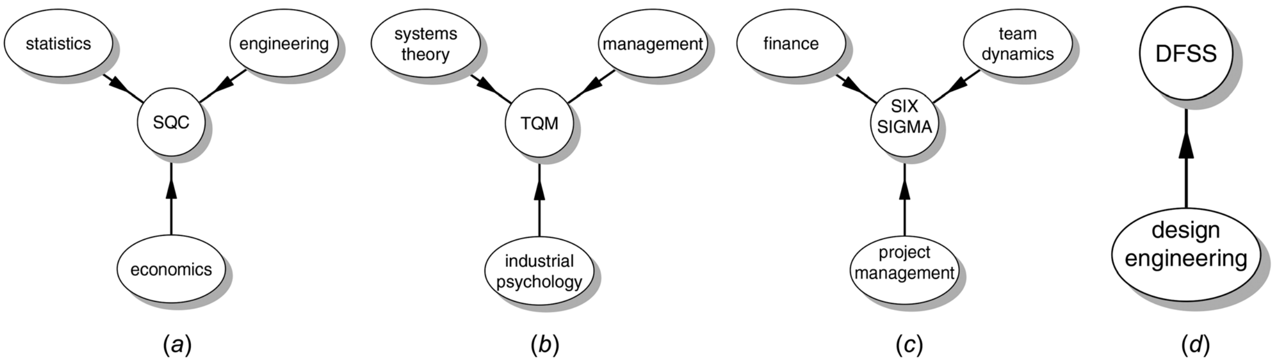
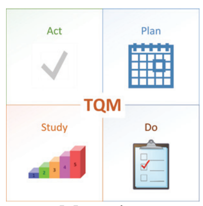
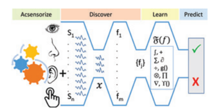


Fig. 18 Quality philosophies: (*a*) statistical quality control, (*b*) total quality management, (*c*) six sigma, and (*d*) design for six sigma

Table 1 Evolution of the problem-solving strategy in the quality movement

Quality philosophy	(a) SQC	(b) TQM	(c) Six sigma	(d) DFSS	(e) PMQ
Quality objective	 Controlling	 Managing	 Reactive	 Proactive	 Predicting
Problem-solving strategy	Specification Production Inspection	Plan Do Check/study Act	Define Measure Analyze Improve Control	Identify Design product Design process Optimize Validate	Acsensorize Discover Learn Predict

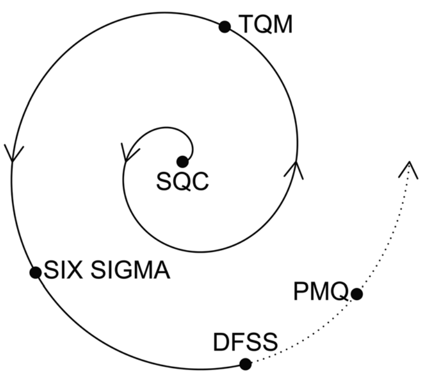


Fig. 19 The quality evolutionary trajectory

matter expertise but no data science background to create empirical models of real, complex processes." In the hands of knowledgeable people, an automated model building system could produce more effective models more efficiently; in the hands of the untrained, it could produce more confusion more quickly. PMQ is a predictive approach that adapted the problem-solving strategy to guide engineers in the data-driven knowledge discovery, for quality control and improvement, Table 1-(e). The effect of PMQ on the quality movement is yet to be determined.

Nomenclature

BD = big data
BDBM = big data—big models
BM = big models
DFSS = design for six sigma
DLL = dynamic-link library
D3M = data-driven discovery of models
GP = genetic programming
ICE = internal combustion engine
LVDT = linear variable differential transformer
MCS = multiple classifier systems
MPCD = maximum probability of correct decision
PMQ = process monitoring for quality
QC = quality control
SQC = statistical quality control
SSP = statistical signal processing
TQM = total quality management
UWBT = ultrasonic welding of battery tabs
α = rate of type I error
β = rate of type II error

References

[1] Harding, J. A., Shahbaz, M., Srinivas, and Kusiak, A., 2006, "Data Mining in Manufacturing: A Review," ASME J. Manuf. Sci. Eng., **128**(4), pp. 969–976.
[2] Cai, W. W., Kang, B., and Hu, S. J., 2017, *Ultrasonic Welding for Lithium-Ion Batteries*, ASME Press, New York.
[3] Zhou, Z.-H., 2012, *Ensemble Methods: Foundations and Algorithms*, CRC Press, Boca Raton, FL.
[4] Breiman, L., 2001, "Random Forests," Mach. Learn., **45**(1), pp. 5–32.
[5] Tulyakov, S., Jaeger, S., Govindaraju, V., and Doermann, D., 2008, "Review of Classifier Combination Methods," *Machine Learning in Document Analysis and Recognition*, Springer, Berlin, pp. 361–386.
[6] Friedman, J., Hastie, T., and Tibshirani, R., 2001, *The Elements of Statistical Learning*, Vol. 1, Statistics Springer, Berlin.
[7] Box, G., 1995, "Total Quality: Its Origins and Its Future," *Total Quality Management*, Springer, Dordrecht, The Netherlands, pp. 119–127.
[8] Astashev, V. K., and Babitsky, V. I., 2007, *Ultrasonic Processes and Machines*, Springer, Berlin.

[9] Kang, B., Cai, W., and Tan, C., 2014, "Dynamic Stress Analysis of Battery Tabs Under Ultrasonic Welding," ASME J. Manuf. Sci. Eng., **136**(4), p. 041011.

[10] Lee, S. S., Kim, T. H., Hu, S. J., Cai, W. W., and Abell, J. A., 2015, "Analysis of Weld Formation in Multilayer Ultrasonic Metal Welding Using High-Speed Images," ASME J. Manuf. Sci. Eng., **137**(3), p. 031016.

[11] Shao, C., Kim, T. H., Hu, S. J., Jin, J. J., Abell, J. A., and Spicer, J. P., 2016, "Tool Wear Monitoring for Ultrasonic Metal Welding of Lithium-Ion Batteries," ASME J. Manuf. Sci. Eng., **138**(5), p. 051005.

[12] Kay, S. M., 1998, *Fundamentals of Statistical Signal Processing, Volume II: Detection Theory* (Prentice Hall Signal Processing Series), Prentice Hall, Upper Saddle River, NJ.

[13] Liu, H., and Motoda, H., 1998, *Feature Extraction, Construction and Selection: A Data Mining Perspective* (The Springer International Series in Engineering and Computer Science), Vol. 453, Springer, New York.

[14] White, H., 2000, "A Reality Check for Data Snooping," Econometrica, **68**(5), pp. 1097–1126.

[15] Leamer, E., 1978, *Specification Searches: Ad Hoc Inference With Nonexperimental Data* (Applied Probability and Statistics), Wiley, New York.

[16] Lee, S., Lee, H., Abbeel, P., and Ng, A., 2006, "Efficient L_1 Regularized Logistic Regression," National Conference on Artificial Intelligence, Boston, MA, July 16–20, Vol. 21, p. 401.

[17] Jensen, D. D., and Cohen, P. R., 2000, "Multiple Comparisons in Induction Algorithms," Mach. Learn., **38**(3), pp. 309–338.

[18] Abell, J. A., Spicer, J. P., Wincek, M. A., Wang, H., and Chakraborty, D., 2014, "Binary Classification of Items of Interest in a Repeatable Process," GM Global Technology Operations LLC, Detroit, MI, U.S. Patent No. US8757469B2.

[19] Dietterich, T., 2000, "Ensemble Methods in Machine Learning," *International Workshop on Multiple Classifier Systems*, Springer, Berlin, pp. 1–15.

[20] Du, S., Liu, C., and Xi, L., 2015, "A Selective Multiclass Support Vector Machine Ensemble Classifier for Engineering Surface Classification Using High Definition Metrology," ASME J. Manuf. Sci. Eng., **137**(1), p. 011003.

[21] Kang, H., and Kim, J., 1997, "A Probabilistic Framework for Combining Multiple Classifiers at Abstract Level," Fourth International Conference on Document Analysis and Recognition (ICDAR), Ulm, Germany, Aug. 18–20, Vol. 2, pp. 870–874.

[22] Ho, T., Hull, J., and Srihari, S., 1994, "Decision Combination in Multiple Classifier Systems," IEEE Trans. Pattern Anal. Mach. Intell., **16**(1), pp. 66–75.

[23] Chakraborty, D., Kovvali, N., Papandreou-Suppappola, A., and Chattopadhyay, A., 2011, "Structural Damage Detection With Insufficient Data Using Transfer Learning Techniques," Proc. SPIE, **7981**, pp. 1–9.

[24] Kang, H., and Kim, J., 1995, "Dependency Relationship Based Decision Combination in Multiple Classifier Systems," 11th International Joint Conference on Artificial Intelligence (IJCAI), Montreal, QC, Canada, Aug. 20–25, Vol. 2, p. 1130.

[25] Chakraborty, D., Kovvali, N., Wei, J., Papandreou-Suppappola, A., Cochran, D., and Chattopadhyay, A., 2009, "Damage Classification Structural Health Monitoring in Bolted Structures Using Time-Frequency Techniques," J. Intell. Mater. Syst. Struct., **20**(11), pp. 1289–1305.

[26] Breivold, H. P., Crnkovic, I., and Larsson, M., 2012, "A Systematic Review of Software Architecture Evolution Research," Inf. Software Technol., **54**(1), pp. 16–40.

[27] Wheeler, D. A., 2003, "Program Library HOWTO," Free Software Foundation, Inc., Boston, MA, accessed June 10, 2017, http://tldp.org/HOWTO/Program-Library-HOWTO/index.html

[28] Spicer, J. P., Chakraborty, D., Wincek, M. A., Wang, H., Abell, J. A., Bracey, J., and Cai, W. W., 2014, "Automatic Monitoring of Vibration Welding Equipment," GM Global Technology Operations LLC, Detroit, MI, U.S. Patent No. US20140138012A1.

[29] Spicer, J. P., Cai, W. W., Chakraborty, D., and Mink, K., 2015, "Clamp Force and Alignment Checking Device," GM Global Technology Operations LLC, Detroit, MI, U.S. Patent No. U.S. 20150165673A1.

[30] Spicer, J. P., Abell, J. A., Wincek, M. A., Chakraborty, D., Bracey, J., Wang, H., Tavora, P. W., Davis, J. S., Hutchinson, D. C., Reardon, R. L., and Utz, S., 2013, "Quality Status Display for a Vibration Welding Process," GM Global Technology Operations LLC, Detroit, MI, U.S. Patent No. US20130105557A1.

[31] Juran, J. M., 1997, "Early SQC: A Historical Supplement," Qual. Prog., **30**(9), pp. 73–81.

[32] Moen, R. D., and Norman, C. L., 2010, "Circling Back," Qual. Prog., **43**(11), pp. 22–28.

[33] Mandru, L., Patrascu, L., Carstea, C.-G., Popesku, A., and Birsan, O., 2011, "Paradigms of Total Quality Management," *Recent Researched in Manufacturing Engineering*, Transilvania University of Braşov, Braşov, Romania, pp. 121–126.

[34] Chua, R., and Janssen, A., 2001, "Six Sigma: A Pursuit of Bottom-Line Results," Eur. Qual., **8**(3), pp. 12–15.

[35] Kwak, Y. H., and Anbari, F. T., 2006, "Benefits, Obstacles, and Future of Six Sigma Approach," Technovation, **26**(5), pp. 708–715.

[36] Chowdhury, S., 2002, *Design for Six Sigma*, Financial Times Prentice Hall, Upper Saddle River, NJ.

[37] Basem, E.-H., 2008, *Design for Six Sigma: A Roadmap for Product Development*, McGraw-Hill, New York.

[38] Shen, W., 2016, "Data-Driven Discovery of Models (D3M)," Defense Advanced Research Projects Agency, Arlington, VA.