Reliable Manufacturing of Heavy Copper Wire Bonds Using Online Parameter Adaptation

Tobias Meyer, Andreas Unger, Simon Althoff, Walter Sextro
Mechatronics and Dynamics
University of Paderborn
33098 Paderborn, Germany
tobias.meyer@uni-paderborn.de

Michael Brökelmann, Matthias Hunstig
Hesse GmbH
Vattmannstr. 6
33100 Paderborn, Germany
broekelmann@hesse-mechatronics.com

Karsten Guth
Infineon Technologies AG
Max-Planck-Straße 5
59581 Warstein, Germany
karsten.guth@infineon.com

***Abstract*—Usage of copper wire bonds allows to push power boundaries imposed by aluminum wire bonds. Copper allows higher electrical, thermal and mechanical loads than aluminum, which currently is the most commonly used material in heavy wire bonding. This is the main driving factor for increased usage of copper in high power applications such as wind turbines, locomotives or electric vehicles. At the same time, usage of copper also increases tool wear and reduces the range of parameter values for a stable process, making the process more challenging. To overcome these drawbacks, parameter adaptation at runtime using self-optimization is desired. A self-optimizing system is based on system objectives that evaluate and quantify system performance. System parameters can be changed at runtime such that pre-selected objective values are reached. For adaptation of bond process parameters, model-based self-optimization is employed. Since it is based on a model of the system, the bond process was modeled. In addition to static model parameters such as wire and substrate material properties and vibration characteristics of transducer and tool, variable model inputs are process parameters. Main simulation result is bonded area in the wire-substrate contact. This model is then used to find valid and optimal working points before operation. The working point is composed of normal force and ultrasonic voltage trajectories, which are usually determined experimentally. Instead, multi-objective optimalization is used to compute trajectories that simultaneously optimize bond quality, process duration, tool wear and probability of tool-substrate contacts. The values of these objectives are computed using the process model. At runtime, selection among pre-determined optimal working points is sufficient to prioritize individual objectives. This way, the computationally expensive process of numerically solving a multiobjective optimal control problem and the demanding high speed bonding process are separated. To evaluate to what extent the pre-defined goals of self-optimization are met, an off-the-shelf heavy wire bonding machine was modified to allow for parameter adaptation and for transmitting of measurement data at runtime. This data is received by an external computer system and evaluated to select a new working point. Then, new process parameters are sent to the modified bonding machine for use for subsequent bonds. With these components, a full self-optimizing system has been implemented.**

***Keywords*-Self-optimization, adaptive system, bond process, copper wire**

I. Introduction

Ultrasonic wire bonding is a well-known process and in industrial use since decades. Due to its favourable properties, it quickly became the interconnection technology of choice for contacting semiconductors. Changes in wire diameter and material demanded changes in the process as well, but it could always fulfill the requirements of new emerging technologies. Today, wire diameter ranges from $15\ \mu\mathrm{m}$ for transmittance of signals all the way to $600\ \mu\mathrm{m}$in the power semiconductor industry. The material change from aluminum to copper in thin and heavy wire bonding prepared the technology for an additional decade.

Regardless of the application at hand, a challenge throughout the years and materials was to handle complexity inherent to wire bonding to estimate the influences of changes of individual single parameters. To cope with the process sensitivity, outer perturbations are reduced to a minimum and bonding parameters that are robust to changes in the wire and substrate are used. They are found with lengthy parameter searches, and once satisfactory parameters are found for a given product, they are supposed to not be changed again. With these, the objectives of the process are also implicitly defined and fixed. Unfortunately, the boundary conditions of the wire bonding process can change and for best results, parameters need to be adapted accordingly. Also a change in the process objectives, e.g. a reduction of the bonding time, must be implemented by changing the process parameters. All these changes would entail a completely new, work-intensive parameter search.

To allow the bonding process to flexibly react to outer changes and to achieve the best result for the defined process objectives, autonomous adaptation of process parameters is required. Self-optimization is one way to implement such behavior adaptation.

A. Structure of this paper

In the following section, general information about self-optimization and the setup of a self-optimizing system is given. Also included is information about multiobjective

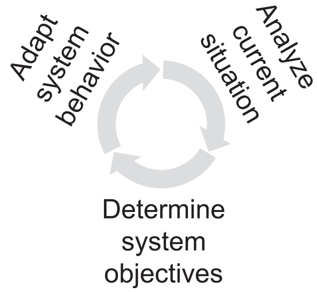

Figure 1. Three steps of the self-optimization cycle.

optimization, which forms a basic prerequisite of self-optimization, and a proposed general structure of information processing. The application of these aspects to the bond machine is given in Sect. III. A main part of it is a detailed discussion of the relevant objectives of the bond process. Section IV explains the changed operational concept, its prototypical implementation and chances for large-scale industrial applications. The last section summarizes the contents of this paper and ends with a brief conclusion.

II. Self-optimization

Self-optimization is a means for implementing autonomous adaptation capabilities in an intelligent mechatronic system [1]. A self-optimizing system is able to adapt its behavior to the current situation and to user demands. The current situation includes changes in environmental conditions but also changes in the system itself, e.g. due to degradation [2].

A great advantage of self-optimization is that it is based on using optimal working points at all times. To facilitate this, system behavior is described in terms of system objectives, i.e. the aspects of system operation that are important to describe overall performance of the system. They need to be quantified, thus forming objective functions, to allow for numerically finding optimal values. A self-optimizing system then pursues its objectives by selecting a trade-off at runtime.

This process makes it possible to separate between offline computation of possible working points and the actual parameter adaptation at runtime, making it uniquely suitable for high-speed applications such as the bonding process.

A. Process of Self-optimization

Self optimization is a constant process comprised of three distinct steps, shown in Fig. 1 [1].

At first, current system behavior is evaluated and actual process objective values are computed. In a next step, currently achieved objective values are compared with desired objective values and a decision is made whether an adaptation of system behavior is required. Lastly, system behavior is adapted. After adaptation, the cycle is closed and system behavior is evaluated again.

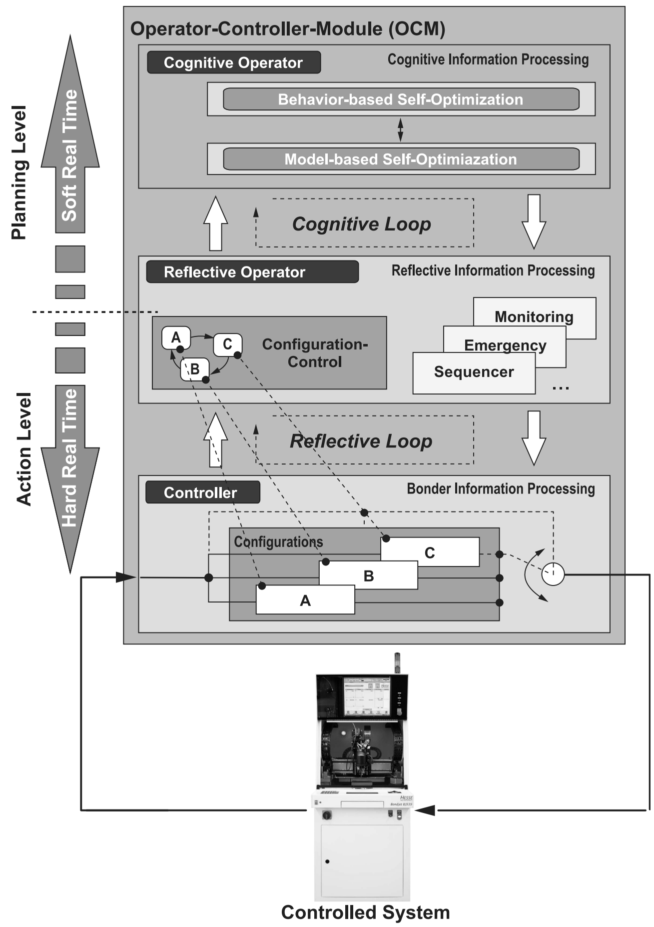

Figure 2. Generic Operator Controller Module ([2])

Not all of the three steps are run in real time on the process control system. Instead, they are implemented as several individual components which are spread over different computer systems throughout the self-optimizing system. This way, each component has access to the ressources it requires – to machine parameters, to process measurements or to high computing power.

B. Operator controller module

For the information processing of a self-optimizing system, the operator controller module was developed [1]. It is based on models of cognitive psychology and is thus related to human behavior, which can also be classified into three disctint levels of consciousness. Its outline is shown in Fig. 2.

The operator controller module contains three distinct levels of information processing. On the lowest level, a real time capable digital controller controls the process itself. It interfaces with sensors to determine the current system state and computes actuator signals such that desired system dynamics are achieved. Interfacing with this controller is the reflective operator which manages desired system behavior and runs emergency routines in case of process failures. It also synchronizes the real time controller with the highest level, the cognitive operator, which operates in soft real time on a slower time scale. The cognitive operator implements selection of objectives and also operating points, and reacts on user input.

To implement self-optimization in a complex system, model-based self-optimization has proven to be a suitable method.

C. Model-based Self-optimization

A basic assumption of a self-optimizing system is that *optimality* of the current working point can be achieved. To this end, objectives are defined and system operation is evaluated with regard to objective fulfillment. However, if the desired trade-off between conflicting objectives is not the current working point, adaptation is required.

Model-based self-optimization has proven to be feasible for different classes of systems [3], [4]. It splits the adaptation process into two separate parts: Computation of optimal working points and choice of working point at runtime.

To compute optimal working points, model-based multiobjective optimization is used. The results then allow a direct choice of the working point.

D. Model-based multiobjective optimization

In optimization, an optimal solution to a nonlinear problem is searched. This problem is usually formulated as minimization problem, i.e. the minimum of a given function is searched. This function is named the *objective function*. If several different objectives are pursued at the same time, the same number of objective functions arise, which are evaluated for the same parameters. In many cases, the minima of individual objective functions occur for different parameter values, creating conflicts among the different objectives. This area between minima contains individual trade-offs, where one objective function value is only decreased at the cost of increasing the value of conflicting objective functions. This set of optimal compromises among objectives is also called *Pareto front*. The set of parameters required for all points of the Pareto front is called *Pareto set*.

Solving a single objective function optimization problem is common and many effective and robust algorithms exist. For multiobjective optimization problems, on the other hand, finding globally optimal solutions is quite difficult. There are two ways: Reduction to a single objective optimization problem or using a dedicated multiobjective optimization algorithm.

For reduction to a single objective, weighted sum, ϵ-constraint and similar methods can be used. All of these come with individual drawbacks that are combined with all drawbacks of the underlying single objective optimization algorithm. They yield one point on the Pareto front per weight, ϵ-constraint, or similar method specific configuration parameters and require a large number of computations in order to find the complete Pareto front. Also, proof that *globally* optimal working points are found is difficult. For these reasons, dedicated multiobjective optimization algorithms are preferred. Examples include genetic algorithms and box subdivision algorithms [5], [6].

In model-based multiobjective optimization, *model* refers to a model of the system that is to be optimized [1]. For self-optimizing systems, a model of system behavior is required. This model has several input parameters, which are used as optimization parameters. All objective functions depend on system behavior. So instead of having several separate objective functions, which are evaluated for the same parameters, parameters are passed to the behavior model and the values of all objective functions are based on a common model simulation. For such complex nonlinear models, analytically determining the gradient of the objective function values with respect to parameters becomes impossible. A strength of box subdivision algorithms is that they need not rely on gradients to find global optima, making them ideally suited for complex behavior optimization tasks [6].

For the self-optimizing bonding machine, the ultrasonic wire bonding process was modelled.

III. Bonding Machine

In the ultrasonic wire bonding machine, model-based self-optimization is implemented. As basis, an off-the-shelf BJ-939 from Hesse Mechatronics was used. The process follows the three steps of a self-optimizing behavior adaptation.

For the second step, determination of desired objectives, model-based multiobjective optimization was employed.

A. Process model

As shown, the basic prerequisite for model-based self-optimization is a complete behavior model. For adaptation of the bonding process, the process itself was modeled.

The general outline of the model was introduced before in [7]. It models the effects of actual process parameters, such as normal force, ultrasonic voltage and process duration, on characteristic properties of the finished bond. It is important to note that process parameters of the model correspond to actual bonding machine process parameters.

As objectives, duration of the process, wear of the tool, tool-substrate contacts and bond strength were determined. Their interdependencies are also shown in Fig. 3 After each process model simulation, objective function values are computed.

1) Duration of the process: Process duration is a direct passthrough from the input parameters, specifically from the duration of voltage and normal force trajectories. A short duration is desired for efficient manufacturing and high

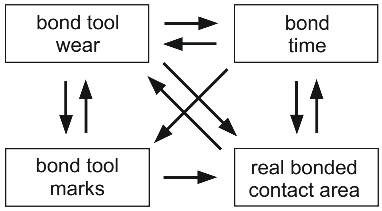

Figure 3. Objectives of the bond process with interdependencies.

machine throughput, but requires higher voltages and forces, which in turn increases the risk of tool-substrate contacts and potentially decreases bond strength.

2) Wear of the tool: The bond tool itself is a major cost driver in copper wire bonding. During manufacturing of the bond, mechanical abrasion leads to tool wear, thus decreasing tool lifetime. To increase usable tool lifetime is a major goal of the self-optimizing bond machine. If absolute tool lifetime was to be used as objective function, simulations of the complete tool lifetime would be required. Simulations run slower than real time, making simulations over the full tool lifetime impossible. Instead, tool wear is computed by including the bond wire in the dynamics model and then computing relative displacement between wire and tool [8]. From this, abrasion for the current bond connection and for currently selected parameters can be determined.

Tool wear occurs mainly in late phases of the wire bonding process. Therefore short process durations can be positive for tool wear, but then strength of the bond is also decreased.

3) Tool-substrate contacts: For good bond quality and high production yield, tool imprints in the substrate, which result from contacts between tool and substrate, have to be avoided. In the model, they are computed from wire deformation. Contacts occur with large wire deformation due to high ultrasonic power, high normal force and long process duration. For large deformations, the probability of contacts increases.

4) Bond strength: Generally, strength of a bond is determined by the mechanical adhesion in the bond interface. Adhesion in the contact area is computed based on the friction energy in the wire-substrate contact area [9].

All four objective function values are based on a common simulation of the process for given process parameters. To ensure model validity for the current process, the estimated state of tool wear from the observer introduced in Sect. III-D2 is incorporated into the model. The main change in the process due to tool wear is in the pressure distribution between wire and substrate, which is important for adhesion between wire and substrate and in consequence for bond strength. In the model, this is computed using finite element models, of which several have been setup for different conditions. The model is parameterized according to the current state of tool wear before being used as basis of the multiobjective optimization.

B. Multiobjective optimization

For the bonding process, optimization parameters correspond to actual process parameters. A box subdivision based multiobjective optimization algorithm was selected for its ability to cope with complex objective functions with unknown gradients, and its ability to make use of parallel computing power, thus reducing computation time. For a given range of all parameters, it evaluates a number of parameter combinations by simulating the bond process model and computing objective function values [6]. Based on these, regions in parameter space that yield unfavorable working points are omitted. Those regions in parameter space where good working points were found are then subdivided into smaller regions and re-evaluted. This iterative process is repeated until a sufficiently large number of feasible working points is found and until each region in parameter space is small enough to be considered *one* working point. Results are objective function values in the Pareto fron and the process parameters required to achieve these in the Pareto set.

During operation, individual objectives can be prioritized over other objectives by selecting desired objective function values. From the Pareto front, the working point closest to the desired objective function values is searched and the corresponding parameters are selected from the Pareto set. The process parameters are then set in the bonding machine to change system behavior.

C. Data transfer

For changing parameters at runtime, Hesse Mechatronics process and production data acquisition system (PBS) is used. Specially adapted bonder firmware requests new parameters in regular intervals and sets these as parameters to be used in subsequent bonds. It allows for passing individual parameters for several coordinate systems. This allows to adapt parameters in one coordinate system while keeping parameters of reference bonds in another system constant. The reference bonds with constant parameters are used for quality evaluations.

For evaluating whether the current working point fulfills current requirements, data from the process is required. For this, Process-integrated Quality Control (PiQC) in the machine is augmented to include full measurement values in addition to quality index values [10]. These are then transmitted via TCP/IP and an external PiQC system to be evaluated in Matlab. From this data, the process observer introduced in Sect. III-D determines the current state of the bonding process and whether a change of operating point is required.

As can easily be deduced, the components of the self-optimizing bonding machine are spread over a multitude of different platforms, which, in combination, form the

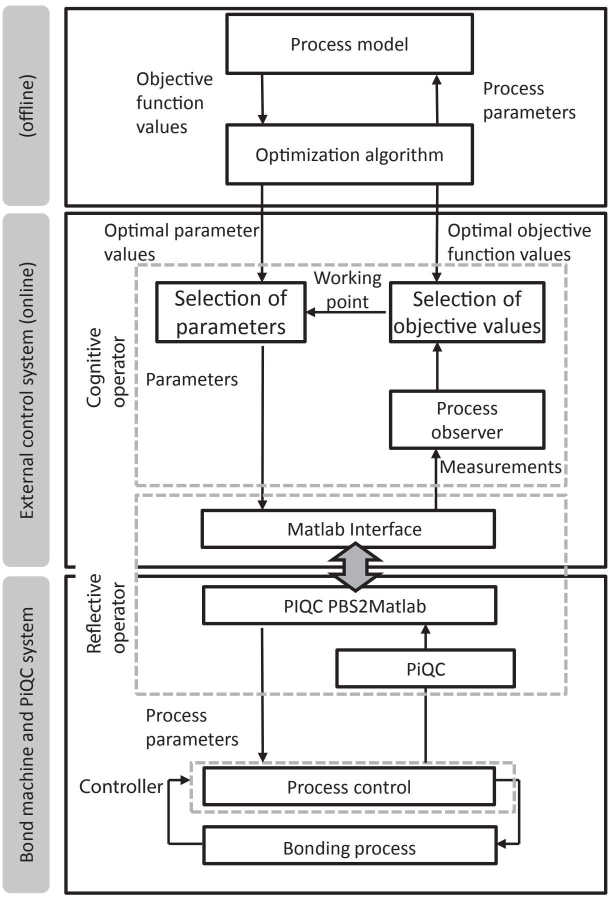

Figure 4. Information processing of the self-optimizing bond machine structured as Operator Controller Module.

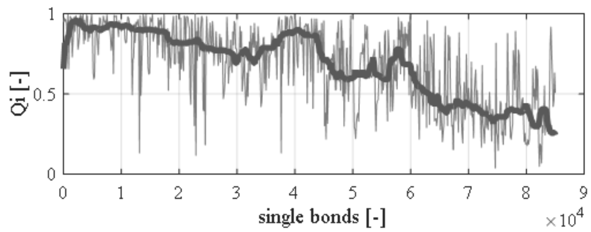

Figure 5. Quality index and derived wear signal.

Operator Controller Module, as shown in Fig. 4. The lowest level is actual machine control and unchanged from a regular bond machine. It works in real time and is capable of closed-loop control of the bond process itself.

The reflective operator is made up of the data interchange structur comprised of machine internal components, PiQC, PBS and an external interface software. Fast, reflective reactions to unexpected events (e.g. premature touchdown detection) are also included in machine control.

The highest level, the cognitive operator, is implemented in Matlab on an external system. Communication with the transitional software is via TCP/IP. For computation of optimal working points to choose from, the University of Paderborn's supercomputer system OCuLUS was used. These computations were done beforehand. Only results, i.e. the Pareto front of optimal objective values and corresponding process parameters in the Pareto set, were used online.

The process data received from the bonding machine is then evaluated using a process observer to determine currently achieved objective function values.

D. Process observer

Information from the actual process about process duration, tool wear, probability of tool-substrate contacts and bond strength needs to be determined to ensure exact operation of the self-optimization.

1) Duration of the process: Information about the duration of the process is directly included in process parameters, mainly in the duration of voltage and force trajectories. Secondary influences, such as movements of the bond head, wire feeding and substrate loading operations, are neglected.

2) Wear of the tool: Estimating the wear rate of the tool is challenging, but there is a range of methods that enable an estimation of the current state of wear at runtime. In the present work a concept of so called quality indices is used. A quality index is a quantitative measurement of the deviation of some sensor data acquired during bonding to previously learned reference data [10]. In this case, the sensor signal of the vertical movement of the tool is sufficiently sensitive to tool wear in order to assess the state of wear [11].

An example of the tool life prediction for a standard copper wire bonding tool made of cermet which has already been in production for 85.000 bonds can be seen in Fig. 5. To compensate for stochastic variation of the signal, a filter operation was integrated. Despite the variation, a reduction in signal value can be observed. In order to use this method, the process parameters need to be constant, which is contradictory to behavior adaptation with self-optimization. To estimate the state of tool wear, dedicated bonds are made in a separate coordinate system which is not affected by parameter adaptation from self-optimization.

To enhance the reliability of this presented method, it is possible to combine results with findings of detected tool-substrate contacts.

3) Tool-substrate contacts: A method for determining the probability of tool-substrate contacts was developed and validated. It is based on finding specific features in the machine signals provided by the PiQC system. This permits the detection of tool-substrate contacts for each bond and the process time at which they occured. During bonding with a worn tool, tool-substrate contacts occur more frequently. This is mainly due to the abrasion of tool material – the V-groove is enlarged, thus allowing more volume of the wire to flow into the opening and resulting in a lowered tool position.

4) Bond strength: A current value of actual bonded area for each bond is required for autonomous behavior adaptation. Currently, an automatic measurement of the actual value of the bonded area is not possible because of the high complexity of the process itself. PiQC allows to determine the strength of each bond made and is nondestructive [10]. Its main result is whether a bond is *similar* to given teach bonds. With this approach, it is limited to detecting deviations from the teach bonds and is thus not suitable for online parameter adaptation, where teach bonds would have to be made for arbitrary process parameters.

Instead, bond strength is used as indicator for the actual objective, bonded area. This can be tested either using shear tests or pull tests. Both values depend on the size of the bonded area and on local adhesion. The currently best-suited test to determine actual bond quality during operation is by using shear tests. In the future it is planned to overcome this problem by using parameter estimation methods with a subset of the partial models required to model the bonding process [7].

All data required for the process observer is continuously generated either during manufacturing of bonds with adapted parameters or in regular intervals between. Both ways, information about those bonds created with arbitrary adapted parameters is obtained.

IV. Operational concept

User-selectable parameters of the self-optimizing bond machine are generalized from actual process parameters to desired process results. These are represented by objective priorities. At the current state of development, they are selected by the machine operator. This detaches her from the process parameters required to achieve the currently desired working point and instead lets her focus on process results.

For the prototypical setup of the self-optimizing bond machine, an external user interface was created in Matlab, shown in Fig. 6. It allows a selection of priorities using graphical sliders, which directly influence the desired objective values. When changing objective priorities, the corresponding working point is selected from the Pareto front and new parameters from the Pareto set are transmitted to the bond machine.

For evaluation purposes, the right side of the graphical user interface also includes process measurements current, wire deformation, and ultrasonic voltage and frequency from all bonds created since the last process adaptation. It allows to switch between reference bonds, which are made for evaluation only, and regular bonds with adapted parameters.

The same computer used for the graphical user interface also runs the high-level information processing including process observer and prompts the operator to input shear test results. Shear tests are conducted on-site to allow for quick response times and quasi-continuous adaptation during

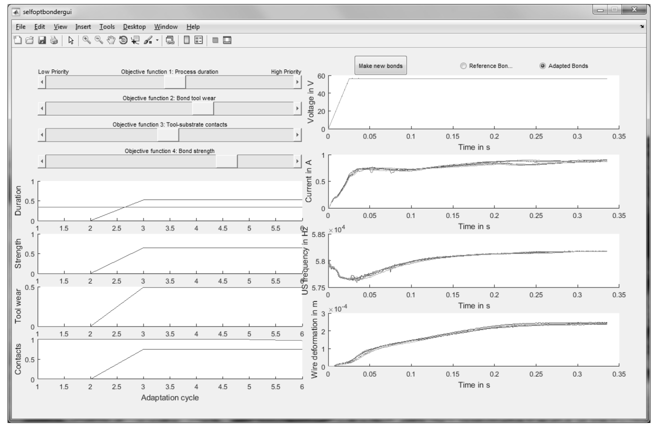

Figure 6. Graphical user interface for control of the self-optimizing bond machine.

experimental validation of the self-optimization process. The process observer determines actual current values of the objectives.

By implementing the high-level intelligence in an external system, only minimal changes were required in machine control. This permits *normal* usage of the laboratory prototype with all options of the current user interface.

For deployment of self-optimizing bond machines on an industrial production scale, additional steps would have to be taken. The general setup with separated high-level intelligence and low level machine adaptation allows processing desired objective priorities on a central production supervision system, where corporate requirements drive the selection of current objective priorities. For example with strict deadlines, cost of high tool wear might be more acceptable than a delay in delivery of products, whereas for normal operation or during a shortage of operating personnel, the supervisor might prioritize tool wear higher and thus reduce the frequency of tool changes.

V. Summary and Conclusion

Heavy copper wire bonding has big advantages like high electrical and thermal conductivity and mechanical stability. But these advantages come at the cost of a process more sensitive towards parameter changes and variations in the environment, e.g. due to tool wear, than standard aluminum wire bonding. It was therefore desired to introduce a self-optimization system into a bonding machine which could compensate such influences.

This was achieved by first setting up a detailed model of the bonding process, which is then used to calculate objective function values for the relevant criteria bond strength, tool wear, tool-substrate contacts and bonding duration. This model is the basis for a multiobjective optimization for finding the best trade-off between the divergent objectives at different states of operation. A special interface enables communication between bonding machine and external control system. It is possible to change machine parameters at runtime as well as to transfer process parameters to the process observer module of the self-optimization system. The four objectives can be prioritized at runtime by the user. The system behavior can be observed in graphs and histograms of the objective functions and in basic process signals.

Within this work it was possible for the first time to implement an adaptive process control system in a self-optimizing wire bonding machine. It was shown that an autonomous reaction to variations in the process environment and to changed production objectives is possible. More investigations using this successful prototype implementation are planned to show its potential as a future new standard in production.

ACKNOWLEDGMENT

This research and development project was funded by the German Federal Ministry of Education and Research (BMBF) within the Leading-Edge Cluster Competition it's OWL (intelligent technical systems OstWestfalenLippe) and managed by the Project Management Agency Karlsruhe (PTKA). The authors are responsible for the contents of this publication.

REFERENCES

[1] J. Gausemeier, F. J. Rammig, and W. Schäfer, Eds., *Design Methodology for Intelligent Technical Systems*, ser. Lecture Notes in Mechanical Engineering. Heidelberg New York Dordrecht London: Springer, 2014.

[2] J. Gausemeier, F. J. Rammig, W. Schäfer, and W. Sextro, Eds., *Dependability of Self-optimizing Mechatronic Systems*, ser. Lecture Notes in Mechanical Engineering. Heidelberg New York Dordrecht London: Springer, 2014.

[3] T. Meyer, C. Hölscher, M. Menke, W. Sextro, and D. Zimmer, "Multiobjective optimization including safety of operation applied to a linear drive system," in *Proc. Appl. Math. Mech.*, vol. 13, 2013, pp. 483–484.

[4] M. Krüger, A. Remirez, J. H. Kessler, and A. Trächtler, "Discrete objective-based control for self-optimizing systems," in *American Control Conference (ACC), 2013*, June 2013, pp. 3403–3408.

[5] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, 2001.

[6] O. Schütze, K. Witting, S. Ober-Blöbaum, and M. Dellnitz, *EVOLVE - A Bridge between Probability, Set Oriented Numerics and Evolutionary Computation*, ser. Studies in Computational Intelligence. Berlin Heidelberg: Springer, 2013, vol. 447, ch. Set Oriented Methods for the Numerical Treatment of Multiobjective Optimization Problems, pp. 187–219. [Online]. Available: http://link.springer.com/book/10.1007/978-3-642-32726-1

[7] T. Meyer, A. Unger, S. Althoff, W. Sextro, M. Brökelmann, M. Hunstig, and K. Guth, "Modeling and simulation of the ultrasonic wire bonding process," in *2015 17th Electronics Packaging Technology Conference*, 2015.

[8] A. Unger, W. Sextro, T. Meyer, P. Eichwald, S. Althoff, F. Eacock, M. Brökelmann, M. Hunstig, and K. Guth, "Modeling of the stick-slip effect in heavy copper wire bonding to determine and reduce tool wear," in *2015 17th Electronics Packaging Technology Conference*, 2015.

[9] S. Althoff, J. Neuhaus, T. Hemsel, and W. Sextro, "Improving the bond quality of copper wire bonds using a friction model approach," in *Electronic Components and Technology Conference (ECTC), 2014 IEEE 64th*, May 2014, pp. 1549–1555.

[10] S. Hagenkötter, M. Brökelmann, and H. J. Hesse, "PiQC - a process integrated quality control for nondestructive evaluation of ultrasonic wire bonds," in *Ultrasonics Symposium, 2008. IUS 2008. IEEE*, Nov 2008, pp. 402–405.

[11] M. Brökelmann, D. Siepe, M. Hunstig, M. McKeown, and K. Oftebro, "Copper wire bonding ready for industrial mass production," *International Symposium on Microelectronics*, vol. 2015, no. 1, pp. 000 399–000 405, 2015. [Online]. Available: http://dx.doi.org/10.4071/isom-2015-WP32