The challenges of statistical experiment design in process development

D. Ortloff* and J. Popp**

Process Relations GmbH
Emil-Figge-Str. 76-80, Germany
44227 Dortmund
*dirk.ortloff@process-relations.com
**jens.popp@process-relations.com

ABSTRACT

Process development, especially in the high-tech sectors, is an extremely complex task. The key challenge is developing a range of process recipes and testing these recipes via experimentation. For setting up the experiments usually a Design of Experiment (DoE) approach is used varying the key driving parameters, running the experiment and evaluating the manufacturing results. Today Statistical Experimental Design (SED) techniques are used to optimize the amount of necessary experiments. But there are challenges with this approach. This paper introduces a software driven development approach covering the full process development cycle. It described how the current practices of SED are extended, matured and how the resilience is improved. It highlights the gains in time and cost. Potential challenges in introducing such an approach are indicated as well.

Keywords: statistical experimental design, process development, development cycle

1 Introduction

Developing high-tech manufacturing processes is a challenging endeavor. It consists mostly of designing a range of process recipes and verifying them in different parameterizations and combinations via simulations and experiments. The high complexity resulting from a huge solution space is especially true for the MEMS industry. This industry is characterized by a large variety and diversity of application and technology domains[1]. Depending on the application a MEMS device may integrate electrical, mechanical, optical, or fluidic components. The diversity of the application requirements and technology options prevents the establishment of a common technology platform like CMOS for the IC-industry. Consequently nearly every MEMS device has a unique device architecture that requires an application specific process for manufacturing. This is typical for MEMS so that it has been coined as the MEMS-Law ("One device, one process") [3]. To execute the verifying experiments can take a lot of time and resources and the gained results are often not conclusive. That again reqiures the definition of further experiments. Additionally, if a recipe contains an error, a whole set of experiments and the used resources like batches of wafers can be wasted, adding significantly to the costs. Experiments to test small variances can all return the same or unwanted results, can be inconclusive or, in the worst case, equipment can be contaminated or damaged by a faulty experiment. As a result even more development delays might occur due to the unavailability of certain equipment.

Over the past twenty years, there has been a focus on shortening process development cycle times. Consequently there has been increasing use of various statistical tools such as 'Statistical Experiment Design' (SED) designed to help automate and accelerate the design process. These techniques use statistical approaches to optimize the set of experiments to execute and therefore reduce the Work-In-Progress (WIP). These statistical approaches have helped to keep up the pace of the ever demanding market, but there are physical limits and different high tech sectors are fast approaching these limits.

Statistical experimental design, especially Fractional Factorial Design (FFD), is a method to significantly enhance experimentation setup and execution. FFD acts as a filter for experiment designs and recipes and uses statistics to expose information about the most important features of the problem studied, while using a fraction of the effort of a full factorial design in terms of experimental runs and resources. In particular FFD can help reduce the number of redundant experiments carried out by development teams, meaning that any experiment that is actually carried out is much more likely to return meaningful results and retrieve meaningful insight and knowledge. This way cause-and-effect chains can be investigated much faster and conclusions can be reached earlier with using less resources.

2 Challenge

Clearly, improving accuracy and WIP in this way is a huge benefit for process engineers - enabling superior results without any extra engineering effort in usually less time. However, there is a major problem with FFD - it is more vulnerable and sensitive to errors. Even the smallest mistake or input error by an engineer can lead to a faulty experiment. The problem with the optimized DoEs is that most or all redundancy is removed from the set of experiments to be executed. On the one hand this is exactly what is desired, on the other hand a single failure, missed result etc. can render the full FFD optimized DoE useless.

In fact one widely respected expert once commented that FFD is "a good way to save a lot of money when done correctly, but is also a good way to burn a lot of wafers". How then do you ensure that SED works every time? Moreover, how can you avoid the waste that process development can generate and save a lot of money? Wafers alone can cost $3000-$5000 each, but significant savings can also be made in relation to time, staff and resource costs.

When investigating the sources of corrupted statistically optimized DoEs two major common reasons can be identified:

- Upfront design errors
- Data collection an evaluation errors

These two sources are described in more detail in the following paragraphs.

Upfront design errors This type of errors results from the increasing complexity. The main challenge for engineers is that new technologies are increasingly complex. At the same time they need to be developed within a much tighter time scale at an increasingly competitive price. The complexity of process development increases every day. More technology options, material choices and supplier options complicate matters more. Therefore the growing length of manufacturing sequences together with more and more material and step interactions make it difficult to comprehend all implications.

Current practice is that senior engineers review newly assembled sequences before they are allowed to be used inside the facilities. This approach prevents many flaws but due the increasing complexity of the manufacturing this task becomes more and more complex and time consuming. Therefore not all potential implications can be spotted all the times, delaying developments due to unsuccessful experiments.

An additional problem is the availability of these experts. There are only very few experts with knowledge of the complete process flow. So they are a bottleneck in the development review process. If the expert is on vacation or unavailable there are only the options to wait or to proceed without review and to risk damaging / contaminating the equipment. Both choices are suboptimal.

Put together these issues can endanger the complete development project. Failed experiments lead to missed budget goals and to delayed product introductions. And these in turn can severely harm the company.

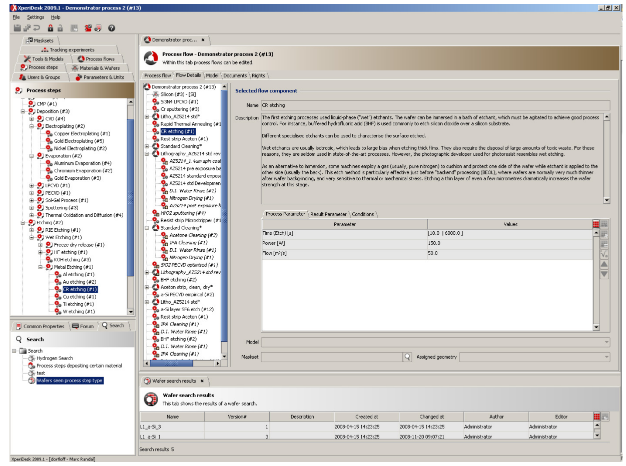

Figure 1: Process library in XperiDesk

Data collection and evaluation errors Another reason why FFD does not always deliver the expected results is the issue of data collection and retention. Experimental results are not always managed in a central location so that it is difficult to see the whole picture. And missing a single result data set can imply to not have the full context of an experiment. This in turn makes the evaluation of the cause and effect chains difficult if not impossible. Therefore the whole DoE might become useless or inconclusive.

Another effect seen from time to time is that experiments not delivering the intended results are not comprehensively documented. Therefore referring back to them somewhere in the future is difficult at best. But the "failed" experiments are exactly the experiments we learn the most from. The experiments delivering the intended results will only tell you parts of the story. On top of that the experiments being a failure in the current development context might be exactly what you are looking for in a future project. Therefore archiving all results comprehensively is key to knowledge generation and retention.

Both error types pose problems that are not simple to solve, as ideally the complete DoE design and evaluation methodology needs to be amended and matured. This can be achieved by extending the currently used software tools with a more holistic approach and software system that needs to cover the recipe design process from start to finish - from verifying the initial design, to optimising the design with SED and finally re-verifying the optimised design.

3 Approach

As mentioned it is only possible to address these issues by applying a comprehensive, software driven approach. Software can store, handle and check large

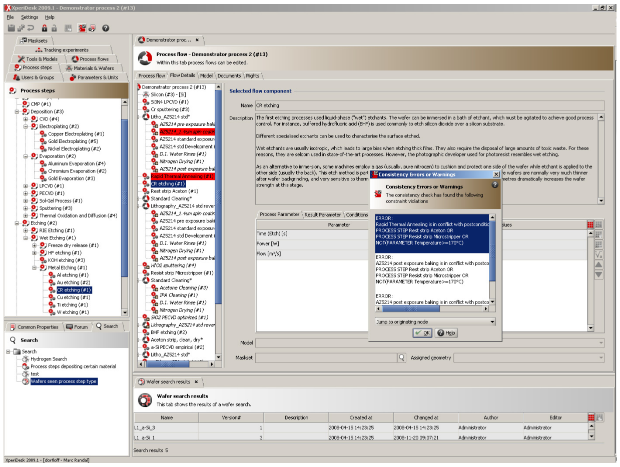

Figure 2: Feasibility and manufacturability check in XperiDesk

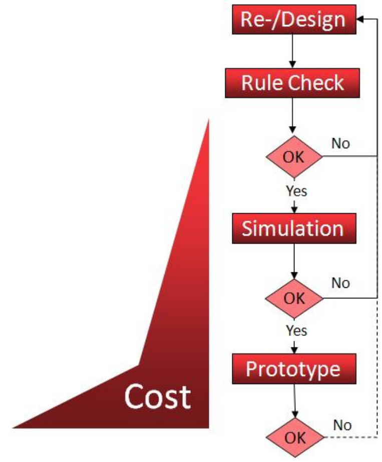

Figure 3: Three phases of process verification

amounts of data in a very short time. The use of software can start in the design phase. Usually development recipes are written in standard office tools. Reuse of processing modules is facilitated by copy-paste functionality. While this method is established it is also very error prone. Corrections of previously done process flows are often only made on paper with a pen. So these corrections are not copied over and those previously made errors are repeated. Software can provide a repository of established process modules (e.g., lithography or cleaning subflows). These can be reused and appended as necessary without damaging future developments. By offering a revision save system the engineer is even able to access previous version and to compare the results and settings. Figure 1 shows such a process library of all process steps categorized into types on the left side. A full manufacturing flow is shown on the right side containing a sequence of manufacturing steps and submodules.

In a next step process flows can be automatically checked in a way introduced in [2]. Since the check of manufacturability is basically checking the process flow for violation of rules (e.g., contamination constraints, parameter constraints, temperature budgets) it can be done fast and efficient with software. A software checker can proof the manufacturability of a new or amended sequence given a set of formalized rules determining a non-functional or non-conformal sequence within seconds. The result of such a manufacturability assessment is presented in figure 2. The figure shows a small inset window indicating rule violations. When selecting one of the contradictions the contradicting process steps are highlighted in red in the process flow. This way engineers can see in a very early stage if a designed process, and all of its branches or splits, is potentially manufacturable.

The check for manufacturability discovers only logical errors in the process flow. It gives no indication of the functionality of the device under construction. A DoE might however generate some experiments that produce non functioning devices. In some cases this can be avoided by applying simulation tools. These tools are however more often than not difficult to use so that engineers spend their time more likely with doing experiments instead of simulating them upfront. By providing an easy interface to these simulators and thus encouraging engineers to simulate the experiments before executing them even more failed experiments can be avoided.

During the optimization of a DoE a software system can then continue to check a design or recipe and the structure at every stage. Potentially harmful or not executable experiments can thus be prevented early on. By using software to verify designs it is possible to continue to extract the benefits of SED in the experimentation while at the same time reducing the associated risk and waste. Figure 3 shows how errors caught early on can reduce the cost of process development significantly.

Additionally this frees a lot of time from the experienced engineers. A first review can now be done by the software and the software might even catch issues that are not easy to spot (e.g., process step 10 indicates a contamination in process step 103). In small and medium sized companies or even in large companies this also limits the risk of loss of knowledge due to personal fluctuation or retirement. Once the rules are formalized, they can be accessed by everybody in the company with the appropriate rights.

Moreover, software can play an even bigger role in the whole process by systematically, automatically and reliably recording the results of experiments. Result data can be automatically captured and linked to the appropriate experiment. New experiments can again

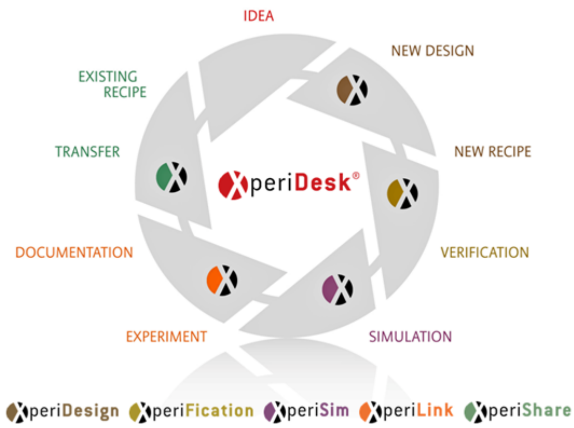

Figure 4: Development cycle

be linked to their projects, wafers, lots, or assessments. The resulting structured data collection enables the process engineer to see the complete picture. An analysis of all captured data from front to back becomes available. Additionally this mechanism collects ALL results; also the results deemed a failure in the current context. Later on these can be searched and might provide a starting point for future development projects. A visioning capability keeping track of all changes and building an audit trail can achieve to build the foundation for regulatory compliance fulfillment.

Software can provide the company with a central development knowledge repository. By using this repository unnecessary experiments can be avoided saving valuable resources and especially time. This software approach helps development teams to properly manage and track all of the inputs and outputs of the whole process around the globe.

4 Conclusions

For high tech process development, software can significantly enhance the optimization and efficiency of the whole design flow. Software is the easiest way to implement complex mathematical and statistical aids in the development flow. The biggest benefit, of course, is that software can remove the burden of tedious and time consuming tasks from engineers - freeing the engineers to concentrate on the more important task of analyzing the data, rather than collecting it. To achieve this it is crucial that the software environment supporting process engineers covers the whole design cycle for new technologies. As depicted in figure 4 this cycle starts with the very idea for a product and ends with the rollout to production. Preventing equipment downtime is another major savings contributor and reducing the Work-in-Progress in the fabrication line by preventing unnecessary experiments through simulation can speed up the developments even more.

REFERENCES

[1] H. van Heeren. Appearance of a moore's law in mems? trends affecting the mnt supply chain. volume 6186. SPIE, 2006.

[2] A. Wagener, J. Popp, T. Schmidt, K. Hahn, R. Brück, and D. Ortloff. Environment for design and verification of mems fabrication processes. In *Proceedings of the MST 2005*, 2005. MST 2005.

[3] Yole Developpement. *MEMS : Global MEMS/Microsystems Markets and Opportunities 2008*. Semi® and Yole Developpement, 2008.