The tough fight for one more bit

Hanspeter Schmid, FHNW/IME
Matlab Expo 2016 Bern

June 23, 2016

I use slides only to illustrate the story I tell, so they do not stand alone. This text re-tells you the story I told at Matlab Expo 2016 and gives you references. It was the last talk of the day.

Today you heard talks about many very interesting applications of MATLAB tools, especially engineering-driven analytics, which we microelectronics designers have been doing for decades now. My talk is not about these big things, though, but about a few small things ...

... microelectronics, yes, but mainly what it means that MATLAB is actually a language and how to use it as such.

CTI Project 7472.2 by FHNW/IME, EPFL/LEG and Colibrys SA was the development of a micro-electromechanical system (MEMS) that measures accelerations. The sensors are etched from silicon wafers, and three wafers are put together to form a tiny sensor element with a mass of only 3 mg hanging on a tiny silicon spring. An acceleration displaces the sensor plate; the location can be read out with one electrical configuration of the sensor, and the sensor plate can be accelerated with another electrical configuration.
Putting the sensor inside a feedback loop gives a servo accelerometer whose plate is well centred. The performance was shown in four publications, Pastre et al. (2009), Schmid et al. (2010), Zwahlen et al. (2010), Pastre et al. (2011). We improved the state of the art by roughly a factor of ten! The disadvantage of closing the feedback loop was that now a wide range of different specialists had influence on the full-system performance and needed to communicate well.

Noise performance graphs like the one shown in the cited papers are very difficult to obtain by measurement and to interpret, so we built a Simulink model (shown in part) in order to facilitate the technical communication in the team. This was necessary because it was impossible to model and predict all effects, and with trial and error we would not get the last bit.

So the team needed to get the last bit by working together, but it was a virtual team with e-mail as its main communication channel. My simplified definition of a virtual team is “a team who do not have coffee breaks together”, and in my experience three things are required to build trust in a virtual team:

- Social exchanges,
- enthusiasm,
- and team members must be able to show they can cope with uncertainty.

The typical way to start a research project, with well-defined work packages and goal-oriented start-up meetings, undermines all three, and if they do not happen on the side, trust is not built!

Even if trust is built, it needs to be maintained, again three things are required:

- Predictable communication (weekly calls, regular meetings, ...),
- fast, substantial responses (the answer “I am quite tied up with other things but will give you the data by next Wednesday” is a substantial, hence trust-maintaining response to the question “Can you send me the data today, please?”; it does not matter that the substance is not the one asked for. “I’ll see what I can do for you” has no substance and no trust-maintaining character.),
- and phlegmatic reactions to crisis (e.g., when the shit hits the fan, hands away from the keyboard, agree to meet in person, and do nothing until then.)
Trust is so important because some thing is always left out when work packages are made, and that thing ends up in a no man’s land. Funnily, this often is full-system performance, because many people believe that the system works if only all the parts work. This is often not sufficient, and then pressure on the teams increases and the no man’s land gets larger because the teams focus more on their core tasks. Then trust is shattered.

Fortunately, the old saying “you never have a second chance to make a first impression” is wrong. You always have one, especially if the early-in-the-project trust building did not happen. In our case, we made a two-day meeting in the Swiss mountains, with half-time project work and half-day work on social exchanges, enthusiasm and coping with uncertainty, much of the latter by playing a poker tournament. (After all, within a project in crisis mode, it is hard to find enough things to be enthusiastic about.) At this time we also established a common Simulink model as one language element of team communication, and all this together enabled us to find that last bit and get the required 19 bits of dynamic range.

So communication is about having a common language, and at FHNW we teach two of them: MATLAB (with Simulink, etc.) and Python (Scientific Python, SciPy). I use a food processor as a metaphor for MATLAB:

- A food processor has lots of very powerful tools,
- the more money you invest, the more tools you get,
- but not everybody has one.

The pocket knife is a metaphor for Scientific Python:

- It is very, very sharp, and efficient,
- it has a limited number of tools, all of them small,
- and everyone has one (or can easily get one; Python is free).

The reader can very easily guess which tool I used to make bread dough, and which one to cut the cucumber. But can you also guess what was done in the following four cases?
We helped Endress+Hauser develop the ASIC and some signal processing inside the ProWirl F200 flow meter under CTI grant 10819.1. Such a flow meter

- combines mechanical sensing, electronics and signal processing,
- needs an interdisciplinary team,
- puts focus on the system.

A clear case for MATLAB, and this was used.

Together with Heliotis, we developed parts of the ASIC and of the signal processing used in their white-light interferometers under CTI grant 14093.1. Such a project

- mainly comprises signal-processing algorithms,
- whose test versions run on FPGAs,
- and whose influences on full-system performance need to be evaluated quickly in order to have shorter design cycles.

This would actually be a case where MATLAB’s HDL coder would come in handy, but Python was chosen as the common language simply because Heliotis always use Python.

With Leica, we worked on sensors used in theodolites under CTI grant 11501.1. There

- the sensor was simulated with field-simulation tools (which neither MATLAB nor Python can do),
- sensor geometry calculations needed to be done,
- and all calculations are vector-based.

Here Scientific Python would be our tool of choice, but MATLAB was used because Leica always uses MATLAB.

Finally, we are also presently working on recommendations on how to do statistical inferences in our field, as shown in Schmid & Huber (2014). This requires

- the development of new algorithms that everyone can read,
- everyone can use and modify,
- and that can be executed for trials with readers’ data on a web server,

http://public.ime.fhnw.ch/threesigma/.

This would anyway be in the domain of the pocket-knife-like general-purpose language Python, but the point that everyone can use it and modify it forces the use of an open-source language.

So mostly we speak MATLAB with MATLAB users and Python with Python users. Occasionally we tell Python users to use MATLAB for one of its tool boxes, or MATLAB users to use Python if a general-purpose mathematical programming language is better suited.
Since Swiss SMEs speak both languages, and the two have very different properties, we also teach the students both languages. This prepares them for a wider range of techno-social interactions.

References


The tough fight for one more bit
Hanspeter Schmid, FHNW/IME.
The tough fight for one more bit
Handling of big data

The future of machine building

The rise of engineering-driven analytics
Handling of big data

The future of machine building

The rise of engineering-driven analytics
big things
small things
important things
Over one million people around the world speak MATLAB.
Engineers and scientists in every field from aerospace and semiconductors to biotech, financial services, and earth and ocean sciences use it to express their ideas.

Do you speak MATLAB?

Saturn's northern latitudes and the moon Mimas.
Image from the Cassini-Huygens mission.
Related article at mathworks.com/ltc
The language of technical computing

Over one million people around the world speak MATLAB. Engineers and scientists in every field from aerospace and semiconductors to biotech, financial services, and earth and ocean sciences use it to express their ideas. Do you speak MATLAB?

Analysis of intraday currency volatility developed by CalPERS. Read more at mathworks.co.uk/finance

Quantitative high-throughput gene expression imaging using data from FlyEx Database. This example available at mathworks.com/ltc


®

©2007 The MathWorks, Inc.

Accelerating the pace of engineering and science
To communicate you need a common language!
Project: MEMS Accelerometer
Locate: measure voltage.
Accelerate: apply field.
System architect
Sensor designer
Analog IC designer
Filter designer
DSP programmer
PCB designer
System architect
Sensor designer
Analog IC designer
Filter designer
DSP programmer
PCB designer
MEMS Accelerometer

19 bits dynamic range

300 Hz bandwidth
MEMS Accelerometer

19 bits dynamic range

300 Hz bandwidth
It is impossible to model everything
Trial and Error:
we reached only 18 bits
The tool alone cannot solve it all
The Team must!
Team
Team: virtual
What is a virtual team?
What is a virtual team?

A virtual team is a team who do not have coffee breaks together!
Building trust early in the project:
Building trust early in the project:

Social exchanges,
Building trust early in the project:

enthusiasm,
Building trust early in the project: coping with uncertainty.
Building trust early in the project:
Maintaining trust later in the project:
Maintaining trust later in the project:

Predictable communication,
Maintaining trust later in the project:

fast, substantial responses,
Maintaining trust later in the project:

phlegmatic reaction to crisis.
Trust is shattered
You never have a second chance
to make a first impression
!!! WRONG !!!
You always have a second chance to make a first impression
You always have a second chance to make a first impression
Off-site meeting in the mountains
Off-site meeting in the mountains

Social exchanges, 
enthusiasm, 
coping with uncertainity.
Off-site meeting in the mountains

POKER NIGHT!
Simulink as a common language
The project became a success:

Four publications;

Communication is about having a common language.
The language of technical computing

Over one million people around the world speak MATLAB. Engineers and scientists in everything from aerospace and semiconductors to biotech, financial services, and earth and ocean sciences use it to express their ideas. Do you speak MATLAB?

Solar Image taken by the X-Ray Telescope: supplied courtesy of Smithsonian Astrophysical Observatory. See related article at mathworks.com/ltc

Accelerating the pace of engineering and science

您会说MATLAB吗？
Language: MATLAB
Language: MATLAB & Python
Introduction

We have all been in this situation: a small number of ICs—some ten or twenty—from a multiproject wafer (MPW) run, and then we are expected to make measurements, derive some statistical data from it, and draw conclusions from the derived data. The simplest way to do statistics is the following.

We assume that we are looking at a small number \( N \) of samples of a larger population. For example, we have \( N = 24 \) test chips from an MPW run. We assume they behave as if they were 24 ICs randomly taken from a huge batch.

We then measure a quantity \( x_i \), getting \( N \) measurement values \( x_i \). For the huge batch, these values have the mean \( \mu \) and the standard deviation \( \sigma \), which we would like to estimate. Without additional prior knowledge, the best estimates are\([1]\):

\[
\text{for } \mu: \quad \bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i; \\
\text{for } \sigma: \quad s = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (x_i - \bar{x})^2}.
\]

An even bigger problem is what it implies when authors plot \( \pm 3\sigma \) limits. Most readers will assume that the \( \pm 3\sigma \) limits drawn are bounds outside, which only 0.27% of the huge batch's samples will be found. This is an even greater fallacy because it is not even correct if we are certain that the values we look at are samples from a normal distribution.

In this article, we will show what is so wrong about using \( \pm 3\sigma \) first, we show what \( \pm \sigma \) and \( \pm 2\sigma \) would really mean if we were certain that we are looking at samples of a normally distributed batch. We are never certain of this, however, and if we measure trimmed ICs, we are even certain that it cannot be a normal distribution. Therefore, we discuss a method of doing statistics that also works if the underlying distribution is not known but has some reasonable shape (being continuous is already more than sufficient).

The main question then remains: How many samples will be outside the limits? We propose a new standard method to define limits that order: \( x_i = -12.237, -9.712, -9.218, -7.235, -6.435, -5.809, -4.884, -4.637, -3.403, -2.527, -1.764, -1.711, -1.613, -0.252, 0.363, 1.109, 1.720, 2.185, 3.750, 5.496, 6.511, 8.722, 10.292, 19.126 \)

To compare different sample sizes, we also look at a second set of data of sample size \( N = 8 \). It is just the first eight ICs that were measured: \( y_i = -7.235, -1.711, 0.363, 3.750, 6.511, 8.722, 10.292, 19.126 \)

We have plotted the data series \( x_i \) as a cumulative distribution in Figure 1: at every value \( x_i \), the curve steps by \( 100/N \). Like this, you can read off the graph, for every \( x_i \), what percentage of the measured points lie below that \( x_i \). We call the point below which \( p \% \) of all points lie \( \hat{F}_p \), the \( p \% \) percentile of the distribution.

\[ P(x_i < \hat{F}_p) = p. \]

The special scaling of the vertical axis would let a normal distribution...
and the 30 Fallacy
Number of Samples
Measuring a Small

Cautions
Swiss SMEs speak both languages
We teach both laguages
Efficient techno-social communication in various, multi-disciplinary projects requires many languages.
Programming languages
natural languages
behavioural languages
Programming languages
natural languages
Team languages
All these are not big things.
These are small things.
But please don’t ever forget:
Care for the small things,
Care for the small things, for in them lies great strength.
The tough fight for one more bit
Care for the small things, for in them lies great strength.
The tough fight for one more bit
Care for the small things, for in them lies great strength.
The tough fight for one more bit.
Thank you for your interest! — hanspeter.schmid@fhnw.ch