A MACHINE LEARNING APPROACH TO THE FREQUENCY CONTROL OF MEMS RESONATORS

Hyung Kyu Lee and Shingo Yoneoka
Stanford University, California, USA

1. INTRODUCTION

Silicon MEMS resonators have been considered as replacements for quartz crystal resonators in electronic systems. MEMS resonators have many advantages over quartz references such as small size, low cost, less power consumption, and CMOS compatibility. Therefore, MEMS resonators are suitable for frequency references (clock) in miniaturized handheld electronic devices.

Silicon MEMS resonators consist of double-ended tuning forks (DETF) that vibrate at a designed frequency when actuation signal is applied (Fig. 1). The resonant frequency is proven to be a function of temperature of the device, dimension of a tuning fork, and the bias voltage(V_{bias}) for actuation. Here, the temperature is environmental variable determined by ambient condition, the geometrical dimensions of a tuning fork are design variables fixed during fabrication, and the bias voltage is a control variable. Since the frequency of silicon MEMS resonators is greatly affected by temperature variation, much worse than that of quartz crystal resonators, frequency stabilization is necessary for MEMS resonators.

The resonant frequency of MEMS resonators can be tuned by changing the bias voltage; therefore, we can stabilize the frequency by applying proper bias voltage according to the measured device temperature. To achieve this goal, we need a calibration table that shows temperature-bias voltage relation for a fixed target frequency. To build this table, we (1) measure frequency at array of measurement points (Fig. 2), (2) fit the data using polynomial equation, (3) and obtain iso-frequency line from the equation.

Conventional calibration method stated above does not care about the efficiency in step (1) and (2). Therefore, a large amount of data is gathered and as many features are used for fitting as possible. However, for the commercialization of MEMS resonators, cost effective calibration process is needed. For that, it is necessary to minimize the number of measurement points to generate a fitting curve without losing the fitting accuracy. Here, a machine learning approach to generate the calibration table would be necessary.

If we figure out significant features and optimized measurement points using machine learning, we can build the same calibration table with shorter measurement time. Also, the device-to-device variation error caused by uncertainties in fabrication process would be solved using the Gaussian process regression algorithm.

For this study, we measure the frequency of multiple resonators as a function of temperature and bias voltage. We will evaluate the performance of our study using this data.

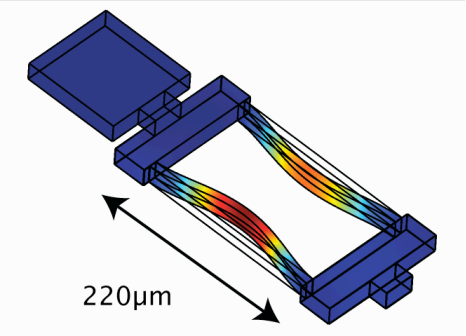


Figure 1: Schematic of the double-ended tuning fork (DETF) MEMS resonator.

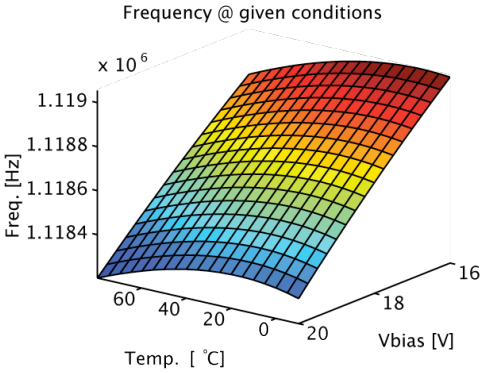


Figure 2: Resonant frequency of MEMS resonator is plotted as a function of temperature and the bias voltage.

2. FEATURE SELECTION

First, we investigate that which polynomial is the best to fit the data for an individual resonator. Hence, we ignore the variation of geometrical dimensions between each device at this point, and assume that the frequency is only the function of temperature and bias voltage. Let's define the temperature as x_1, and a bias voltage as x_2. We have m training examples $\{x^{(i)}, f^{(i)}: i = 1,2, \dots, m\}$, where $x^{(i)} = [x_1^{(i)}, x_2^{(i)}]^T$ and $f^{(i)}$ is measured frequency. The fitting curve for the frequency of the resonator can be described as a combination of polynomials in two variables.

$$f^{(i)} = \sum_j \sum_k a_{(j,k)} (x_1^{(i)})^j (x_2^{(i)})^k \quad (1)$$
$$where, 0 \le j, 0 \le k, i + j \le n$$

If we treat all variables as high dimensional input features such that $z = [x_1^n, x_2^n, x_1^{n-1}x_2, x_1x_2^{n-1}, \dots, x_1, x_2, 1]^T$ then, we can linearized the equation as $f = z^T\theta$. θ can be calculated with a normal equation.

$$\theta = (\tilde{Z}^T\tilde{Z})^{-1}\tilde{Z}^T\vec{f}, \qquad \vec{f} = \begin{bmatrix} f^{(1)} \\ \vdots \\ f^{(m)} \end{bmatrix} \qquad \tilde{Z} = \begin{bmatrix} z^{(1)T} \\ \vdots \\ z^{(m)T} \end{bmatrix}$$

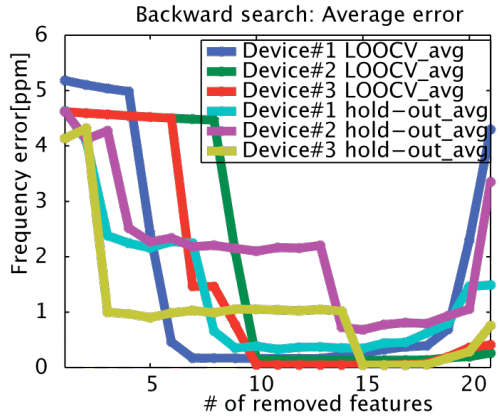
Since we have a large number of features(28 features for n =6), we conduct feature selection to extract important features using backward and forward search. We use hold-out cross validation (70% of the data) and leave-one-out cross validation (LOOCV) to calculate the generalization error for the feature selections. We use two criteria in cross validation routine; 'maximum error method (MaxE)' and 'average error method (AvgE)'. In each iteration step, we obtain the array of error values correspond to learning examples. Here, MaxE uses the maximum value of errors and AvgE uses the averaged value to choose more significant feature set. We start feature selection from the 6th order-2 variable polynomial.

Fig. 3 shows the result of feature selections for three different devices. We compared the number of removed features with the error for the backward search (Fig. 3 (i, ii)), and the number of features with the error for the forward search (Fig. 3 (iii, iv)). AvgE and MaxE are examined in each case.

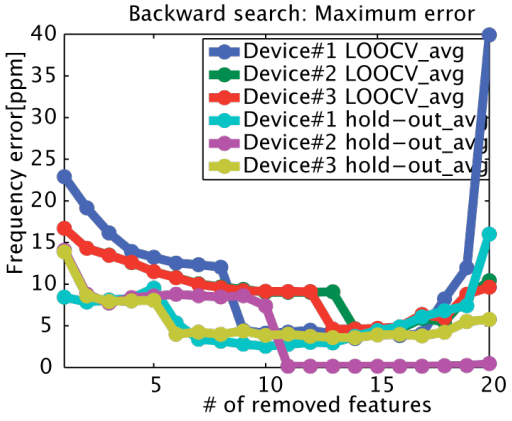
From the calculation result, we can observe that the average error is minimized when the number of removed features is 13 – 15 when we apply backward search. For the forward search, the average error is minimized when the number of feature is 10 – 15. Larger 'number of removed features' is preferred for backward search and smaller 'number of features' for forward search since we want to reduce the measurement points to generate the fitting curve. Using the measurement data of those devices, we summarize top 14 features to be removed or chosen by each search algorithm(*Table 1*).

From the analysis of the results from both search methods, we found that we can obtain the best fitting error when the number of feature is ~13. We conducted feature selection for the data from three different devices, and obtain similar trend for the optimized number of features. Therefore, we conclude that the numbers of significant features are common for different devices.

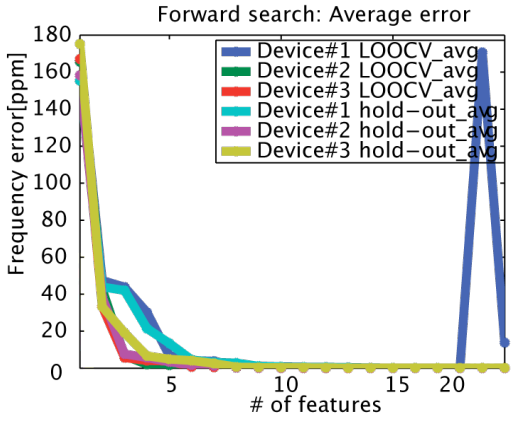
(i)



(ii)



(iii)



(iv)

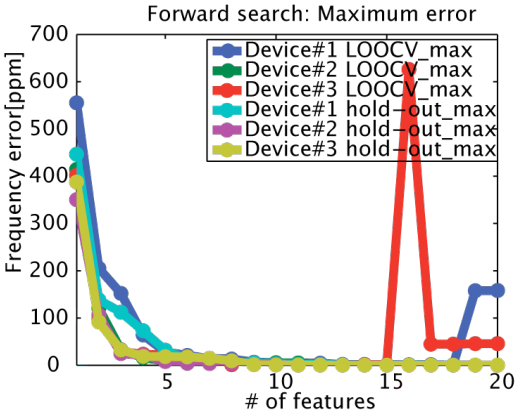


Figure 3: Calculation result form feature selection algorithm. (i) and (ii) show the average and the maximum error for backward search. (iii) and (iv) show the average and the maximum error for forward search. 2 different types of cross validation methods are used to evaluate errors.

Table 1: Top 14-feature list chosen by feature selection algorithm.

Backward	AvgE
LOOCV	$x_1^6, x_1^3x_2^3, x_1^3, x_1^4x_2^2, x_1^5x_2, x_1^4x_2, x_2, x_1^3x_2^2, x_1^5, x_2^6, x_1, x_1^2, x_2^4, x_1x_2^5$
Hold out	$x_1^6, x_1^5x_2, x_1^2x_2, x_1^5, x_1^4x_2, x_1^3x_2^3, x_2^2, x_1^2x_2^3, x_1^3x_2^2, x_1x_2^5, x_1^4, x_2, x_1^3x_2, x_1x_2$

Backward	MaxE
LOOCV	$x_1^6, x_1^3x_2, x_1^2, x_1^2x_2, x_1^2x_2^4, x_1^3, x_1^2x_2^3, x_1^3x_2^3, x_1x_2^5, x_1^3x_2^2, x_1^4x_2^2$, *const*, $x_1x_2^3, x_1$
Hold out	$x_1^6, x_1^4x_2, x_1^3x_2, x_1x_2^5, x_1^4, x_1^5, x_1^4x_2^2, x_1^5x_2, x_1^3x_2^2, x_1^2x_2, x_1^3, x_1x_2^2, x_1x_2, x_1x_2^4$

Forward	AvgE
LOOCV	*const*, $x_1^2, x_2^2, x_1^4x_2^2, x_1x_2^5, x_1^4x_2^1, x_1^2x_2^3, x_1^3, x_1^5x_2, x_1^3x_2^2, x_2^6, x_1, x_1^5, x_1^6$
Hold out	*const*, $x_2, x_1^3x_2^2, x_1^2, x_1x_2^5, x_1^4x_2^1, x_1^3x_2^1, x_1^3, x_1^2x_2, x_1^5, x_1x_2^3, x_1^2x_2^3, x_1x_2^2, x_2^6$

Forward	MaxE
LOOCV	*const*, $x_1x_2^5, x_2^3, x_1^2, x_1^3, x_2^5, x_2, x_2^6, x_1^4x_2, x_1, x_1^4, x_1^6, x_1^2x_2^4, x_1^3x_2^2$
Hold out	*const*, $x_1x_2^4, x_1^5, x_1x_2, x_2, x_1^3x_2, x_2^2, x_1^2x_2^2, x_1^4x_2^2, x_1^3, x_2^5, x_1x_2^5, x_1^4x_2, x_1$

Random search for feature selection

From the previous section, we decided how many and which features are needed for the fit with the smallest fitting error. However, feature selection methods (forward search and backward search) we used above are heuristic so that they do not guarantee that the solution has the minimum fitting error. It becomes more evident if we see solutions presented on Table 1. Although there are some correlations among solutions, no unique solution exists. Therefore, we can conclude that there is room for improvement with other search method. This is the motivation to implement the 'random search' method for feature selection.

The random search method compares fitting error of a large number of possible feature sets in a loop and returns the feature set with the smallest error. At the first step, randomized feature set is generated by the code. Since we start from 6th order-2 variable polynomial equation, we have total of 28 features in the feature pool. Therefore, if we want to select N features through random search algorithm, we have ${}_{28}C_N$ different features sets to compare. Here, we will use N=13 because we learned 13 is the minimum number of features with which we can achieve sub-ppm error. Hence, at least ${}_{28}C_{13}$=37,442,260 iterations are needed to obtain the solution in an ideal case. This is computationally heavy process so that couple of weeks is required on a personal computer. Therefore, we decided to run a routine until top 10 smallest errors are smaller than backward / forward search result, and then analyze feature sets in a top 10 list. This modification makes the result of random search method not to be the guaranteed best solution, but pseudo best solutions will be still meaningful if we can improve fitting error.

In the implementation, device #1 and #2 are used with 'LOOCV' for cross validation and MaxE for error criteria. After 1 million iterations, results are compared to forward search results (Table 2). If we obtain the most frequently appeared features from the top 10 feature set list, we can build a feature set with small fitting error. The resultant feature set is shown in Table 3.

Table 2: Comparison of fitting errors between forward search and random search.

Device	Forward search	Random search: top 10
#1	0.6712 ppm	0.5462 to 0.6070 ppm
#2	0.1473 ppm	0.0941 to 0.1149 ppm

Comparison with analytical model

Finally, we compare the result of feature selection from random search with the analytical model based on the physics of the resonant beam. The frequency of the resonant beam is a function of the temperature and the bias voltage and it can be described as following in a simple model:

$$f(T,V_b)=\frac{1}{2\pi\sqrt{\rho_{Si}(T)A_c(T)}}\sqrt{\frac{22.4^2}{l(T)^4}B(T)-\frac{2h_{Si}(T)\varepsilon_0\varepsilon^3{}_{SiO_2}V_b^2}{(2t_{SiO_2}(T)\varepsilon_0+g_{beam}(T)\varepsilon_{SiO_2})^3}} \quad (2)$$

where ρ_{Si} is the mass density of silicon, A_C is the cross-sectional area of the beam, l is the length, B is the bending stiffness, h_{Si} is the height, t_{SiO2} is the thickness of SiO_2 layer, and g_{beam} is the gap for electrostatic transduction. These material and dimensional variables are known to be a function of temperature. ε_0 and ε_{Si} are permittivity constants.

We conduct Taylor expansion of the Eqn. 2 to express frequency as a polynomial function of temperature and bias voltage to make Eqn. 2 comparable to Eqn. 1. After Taylor expansion, we compare which terms in the resultant polynomials are important by checking variance of each terms in the given temperature and bias voltage range.

Table 3 shows the comparison of important features from the analytical model approach and the random search method. Features in red color are common features from those two different methods. From this result, we observe that 7 out of 13 features are shared. This is slightly more than 50%, which implies that our feature selection methods are reasonable in physical sense. However, there are still mismatch of features. It suggests the current analytical model is not sophisticated enough to describe the relation between frequency, temperature, and the bias voltage correctly. This is true because a known phenomenon such as 'A-f effect' is not included in Eqn. 2 due to its complexity. Also, it is possible that unknown factors affect the measurement result, causing Eqn. 2 not to predict frequency precisely.

In conclusion, comparison between analytical model and results from feature selection proved the reliability of feature selection method we used. Also, it gives us the need to improve the analytical model for better prediction.

Table 3: Comparison of the combination of 13 best features extracted by random feature selection algorithm and the analytical mode.

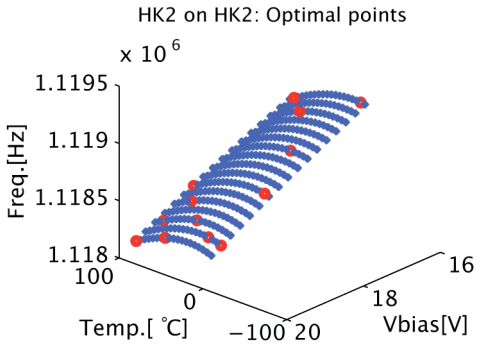
Random feature selection	const., $x_1^3x_2^3$, x_1^2, x_1^5, x_2^2, $x_1x_2^3$, $x_1^2x_2^4$, $x_1x_2^4$, x_1, $x_1^4x_2^2$, x_2^5, $x_1^2x_2^3$, x_1^4
Analytical model	const., x_2^2, x_2^4, x_1^2, x_1, $x_1^1x_2^2$, x_2^6, $x_1^2x_2^2$, $x_1x_2^4$, x_2^5, $x_1^2x_2^4$, $x_1x_2^6$, x_2^3

2. OPTIMIZED CALIBRATION POINT

In the previous chapter, we could specify the combination of optimized features using feature selection. As a next step, we minimize the number of measurement points to generate calibration tables with similar accuracy. Since the optimized number of feature is 13, we only need 13 measurement points to generate calibration table. We reduce unimportant measurement points using backward search algorithm. Fig. 4 show the fitting curve which is generated by 13 measurement points chosen by backward search algorithm, and the error between the fitting curve and the measurement values. The maximum error of the fitting curve is 0.520 ppm. Since we originally need 323 points to generate the calibration table with 0.09 ppm maximum error, we could achieve 25 times reduction of measurement points with maintaining sub ppm error.

Optimized measurement points chosen by a certain device should be applicable for other devices. Hence, we generate fitting curve for other device using the same measurement point. Comparison of maximum and mean error of each device is shown in Table 3. We can see that all maximum error is sub-ppm.

(i)



(ii)

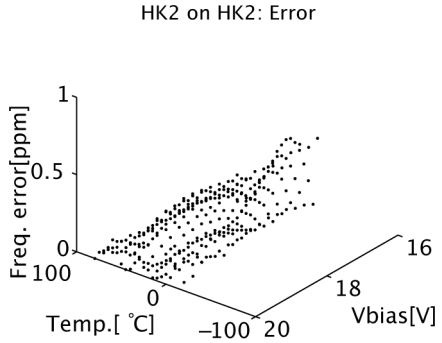


Figure 4: (i) Calibration table generated by the optimized 13 measurements point using backward search. (ii) The error between the generated calibration table and measurement result.

Table 3: Maximum and mean errors of calibration table generated by 13 measurement points. Measurement point is specified by "original" device with backward search algorithm, and those points are used for the calibration in device # 1, 2, and 3.

Device #	Original	1	2	3
Max. error [ppm]	0.520	0.797	0.302	0.290
Mean error [ppm]	0.0965	0.139	0.0701	0.0710

3. DEVICE-TO-DEVICE VARIATION

In previous two parts, we figured out the way to obtain a calibration curve for a device in efficient way. Although those are great accomplishment, we want to further reduce the number of measurement points.

The idea starts from the fact that resonators with the same design share general frequency-temperature-bias voltage characteristics even if there is a small variation due to uncertainties in manufacturing process. In other words, although we cannot apply the characterization result of one device to other devices directly, it is possible to estimate the fitting curve of a test device if we combine full characterization data of training devices with a few measurement data of the test device.

Since the variation is caused by latent (unobservable) variables such as dimensional variables of the resonant beam, we need additional variables on top of temperature and bias voltage to learn general characteristics. The frequency of at certain temperature and bias voltage points are only option for this purpose because there are no more observable variables.

Unlike Eqn. 1, which is a good hypothesis for a single resonator characteristic, we don't know which hypothesis effective to treat device-to-device variation due to additional input variables. Therefore, we decide to use the Gaussian process regression since it free us from the difficulty to choose right hypothesis.

Let's define training input as X_{IN}, training output as F, testing input as X^*_{IN}, and the testing output as F^*. As additional input variables to identify each device and to learn general characteristic, we include the frequency of certain measurement points to input X_{IN}. If we define:

m: # of on calibration data on each device
n: # of devices for training
k: # of measurement data on a test device

Input for the Gaussian process regression can be described as:

$$F = \begin{bmatrix} \vec{f}_1 \\ \vec{f}_2 \\ \vdots \\ \vec{f}_n \\ \vec{f}_t \end{bmatrix} \in R^{(nm+k)\times 1}, \quad X_{IN} = \begin{bmatrix} X_1 & Y_1 \\ \vdots & \vdots \\ X_n & Y_n \\ X_t & Y_t \end{bmatrix} \in R^{(nm+k)\times(k+2)}$$

$$F^* = \begin{bmatrix} \vec{f}_{t^*} \end{bmatrix} \in R^{(m-k)\times 1}, \quad X^*_{IN} = \begin{bmatrix} X_{t^*} & Y_{t^*} \end{bmatrix} \in R^{(m-k)\times(k+2)}$$

where,

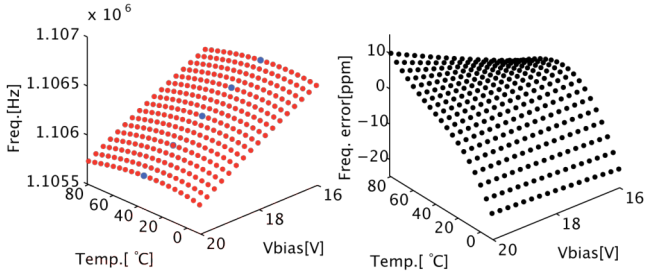
$$\vec{f}_i = \begin{bmatrix} f_i(x^{(1)}) \\ f_i(x^{(2)}) \\ \vdots \\ f_i(x^{(m)}) \end{bmatrix}, \; X_i = \begin{bmatrix} x_{i,1}^{(1)} & x_{i,2}^{(1)} \\ x_{i,1}^{(2)} & x_{i,2}^{(1)} \\ \vdots & \vdots \\ x_{i,1}^{(m)} & x_{i,2}^{(m)} \end{bmatrix}, \; Y_i = \begin{bmatrix} f_{i,1} & \cdots & f_{i,k} \\ f_{i,1} & & \\ \vdots & & \vdots \\ f_{i,1} & \cdots & f_{i,k} \end{bmatrix}$$

$$\vec{f}_t = \begin{bmatrix} f_t(x^{(1)}) \\ f_t(x^{(2)}) \\ \vdots \\ f_t(x^{(k)}) \end{bmatrix}, \; X_t = \begin{bmatrix} x_{t,1}^{(1)} & x_{t,2}^{(1)} \\ x_{t,1}^{(2)} & x_{t,2}^{(2)} \\ \vdots & \vdots \\ x_{t,1}^{(k)} & x_{t,2}^{(k)} \end{bmatrix}, \; Y_t = \begin{bmatrix} f_{t,1} & \cdots & f_{t,k} \\ f_{t,1} & & f_{t,k} \\ \vdots & & \vdots \\ f_{t,1} & \cdots & f_{t,k} \end{bmatrix}$$

$$\vec{f}_{t^*} = \begin{bmatrix} f_t(x^{(k+1)}) \\ f_t(x^{(k+2)}) \\ \vdots \\ f_t(x^{(m)}) \end{bmatrix}, \; X_{t^*} = \begin{bmatrix} x_{t,1}^{(k+1)} & x_{t,2}^{(k+1)} \\ x_{t,1}^{(k+2)} & x_{t,2}^{(k+2)} \\ \vdots & \vdots \\ x_{t,1}^{(m)} & x_{t,2}^{(m)} \end{bmatrix}, \; Y_{t^*} = \begin{bmatrix} f_{t,1} & \cdots & f_{t,k} \\ f_{t,1} & & f_{t,k} \\ \vdots & & \vdots \\ f_{t,1} & \cdots & f_{t,k} \end{bmatrix}$$

In the implementation, we prepare measurement data of 3 devices with identical design. Then, we use full characterization data of 2 devices with data from a few measurement point of the test device for training with the squared exponential kernel. After optimizing parameters for the regression, we generate the calibration table (Fig. 5). Less than 20 ppm error is achieved with 5 points measurement, and less than 40 ppm error with 1 point measurement. This is great result considering we use only two devices for training. Further improvement is expected in the future when more training devices are prepared.

(i)



(ii)

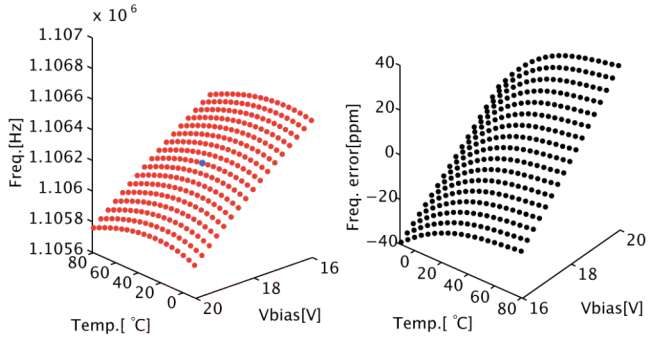


Figure 5: Calibration table generated by (i) 5 point measurements and (ii) 1 point measurement based on the training data of 2 devices.

CONCLUSION

Using tools in machine learning, we success to develop effective calibration process for the frequency control of MEMS resonators. We achieve 25 times reduction of measurement points for the calibration with maintaining sub-ppm error between the fitting curve and measurement values. Also, we demonstrate the use of Gaussian process regression to solve device-to-device variation issues. Since the current calibration error in this part is not sufficient for the commercialization, future work would be focused on the improvement of this problem using large amount of training data.

ACKNOWLEDGEMENT

We would like to thank Prof. Ng and CS229 teaching team for great learning experiences.

REFERENCES

[1] CS229 Machine Learning, Lecture Note

[2] S. Chatterjee and A. S. Hadi, *Regression Analysis by Example*. Hoboken: John Wiley & Sons, 2006.

[3] C. E. Rasmussen and C. K. I. Williams, "*Gaussian Process for Machine Learning,"* MIT Press, 2006.

[4] C. T. C. Nguyen, "*MEMS technology for timing and frequency control*," IEEE transactions on ultrasonics, ferroelectrics, and frequency control, vol. 54, pp. 251-70 (2007).

[5] G. K. Ho, K. Sundaresan, S. Pourkamali, and F. Ayazi, "*Temperature compensated IBAR reference oscillators*," in 19th IEEE International Conference on Micro Electro Mechanical Systems (MEMS'06), Istanbul, Turkey (2006), pp. 910-913.

[6] R. Melamud, B. Kim, S. Chandorkar , M. A. Hopcroft, M. Agarwal, C. M. Jha, and T. W. Kenny, "*Temperature-compensated high-stability silicon resonators*," Applied physics letters, vol. 90, 244107 (2007).

[7] http://www.vectron.com/products/tcxo/tcxo_index.htm