

## 面状振動する超音波振動源に用いる共振追尾のシミュレーション

### Simulating a Resonance-Tracking for Ultrasonic Vibration Source Generates Planar Vibration Locus

○青木耀佑<sup>1</sup>, 三浦光<sup>2</sup>, 浅見拓哉<sup>2</sup>

\*Yosuke Aoki<sup>1</sup>, Hikaru Miura<sup>2</sup>, Takuya Asami<sup>2</sup>

Abstract: We have developed a resonance-tracking program for an ultrasonic vibration source, which have two resonance frequencies. In this study, we simulated the resonance-tracking program by using Simulink. As a result, resonance-tracking was completed in 1 ms.

#### 1. はじめに

ワイヤボンディングにおいて、2次元振動である面状振動による超音波接合は、従来の1次元振動である直線振動による接合に比べ、単位時間毎の接合強度が増加する可能性がある。そこで、筆者らは2つの周波数で駆動させることで面状振動する超音波振動源<sup>[1]</sup>を対象とした共振追尾回路を開発している。本検討では2つの共振周波数を持つモデルに対し共振追尾のシミュレーションをSimulinkにて行った。

#### 2. 共振追尾

共振追尾とは、振動源への入力信号の周波数を振動源固有の共振周波数と一致させることであり、これにより振動源は継続的に効率よく振動する。本検討では、非共振時に生じる電圧と電流の位相差を基に入力信号の周波数を増減させて共振周波数へ近づける。Fig. 1は、本稿で検討した共振追尾のブロック図である。図に示す通り、面状振動する振動源では、振動源が持つ2つの共振周波数に対して共振追尾を行う。したがって、振動源の出力は2つ周波数成分を持つ歪み波であるため、フィルタでそれぞれ単一周波数の信号に分割する。信号発生部では、まず同じ周波数の電圧  $V_1$  と  $V_2$ 、及び電流  $I_1$  と  $I_2$  からそれぞれ電圧と電流の位相差  $\theta_1$  と  $\theta_2$  を求める。次にこの位相差を基に現在の周波数  $f_1$  と  $f_2$  を増減させ、新たな周波数の入力電圧  $V(f_1)$  と  $V(f_2)$  を発生させる。

#### 3. シミュレーションと結果

振動源は、2つの共振周波数を  $F_1=60$  kHz と  $F_2=130$  kHz に定め、それぞれを共振周波数とする RLC 直列回路を並列に接続した回路にモデル化した。また、 $F_1$  及び  $F_2$  に対する入力電圧の初期周波数はそれぞれ  $f_{1-0}=50$  kHz,  $f_{2-0}=120$  kHz とした。

その結果として、シミュレート時間に対する入力電圧の周波数の変化を Fig.2 に示す。図より、 $F_1$  を目標

とする入力電圧の周波数  $f_1$  は 60 kHz でほぼ一定となり、 $F_2$  を目標とする入力電圧の周波数  $f_2$  は 130 kHz でほぼ一定となった。よって、 $f_1$  及び  $f_2$  はそれぞれの目標とする共振周波数  $F_1$  及び  $F_2$  に共振追尾が出来ていることが分かった。また、 $f_1$  と  $f_2$  の共振追尾は共に 1 ms 時点で完了することが分かった。

#### 4. おわりに

本検討では、2次元の振動軌跡となる超音波振動源を2つの共振周波数をもつRLC回路にモデル化し、共振追尾のシミュレーションを行った。結果、初期の入力周波数が共振周波数より約 10 kHz 小さい場合に、2つの共振周波数に対しそれぞれ約 1 ms で共振追尾が完了するモデルを作成することができた。

#### 参考文献

[1] Y. Tamada, T. Asami and H. Miura, Acoust. Sci. Tech., 41 (4), 645-653, 2020.

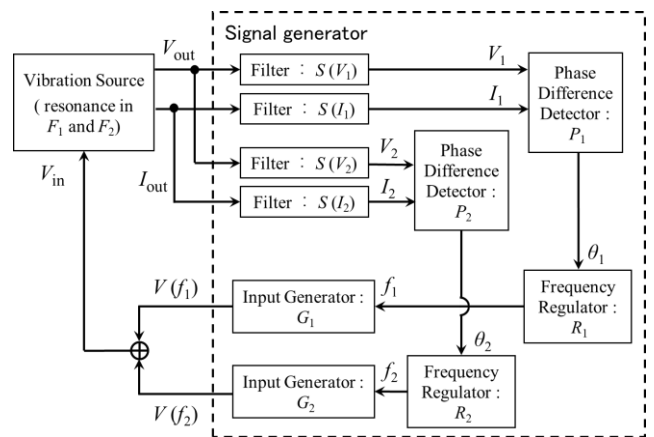


Fig. 1 Block diagram of resonance-tracking.

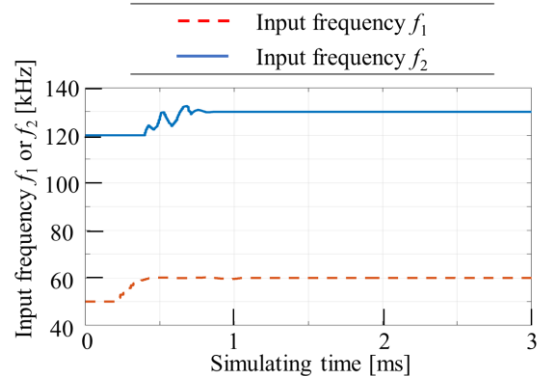


Fig. 2 Input frequency vs. simulating time.