

# A Parallel Simulation Code for Synchronization of Spin Torque Oscillators for Microwave Applications

Yan Zhou<sup>+</sup>, Ada Zheng, and Herman Wu

Department of Information Technology, Hong Kong Institute of Technology, 2 Breezy Path, Mid-levels West, Hong Kong

**Abstract.** A parallel Fortran 90 code based on MPI is developed to study the synchronization of coupled spin torque oscillator (STO) network for applications in communication technology. The code is successfully implemented on the Dell Xeon cluster with a total of 442 nodes and simulations of up to 1600 serially connected STOs are carried out, which to date is the most STOs ever simulated. The performance of the MPI-based parallel code has been benchmarked and discussed.

**Keywords:** Spin torque oscillators, MPI, parallel efficiency

## 1. Introduction

Compared to the dominant Si-based CMOS technology based on the charge information carrier, spintronics nano-devices rely on not only the electron's charge but also its spin and is thus believed to be an alternative technology to overcome the hurdle of heat generating and leakage currents on the roadmap for 22nm CMOS logic and beyond [1,2]. Based on the giant magneto resistance (GMR) and spin momentum transfer effect – the magnetization orientation change will cause changes in conductance of the system or vice versa, spintronics devices have been widely used in today's high capacity hard drives and magnetic recording industry [3,4,5]. Recently, tremendous research and development efforts have been devoted to the following two most promising spintronics technologies – spin torque oscillators (STOs) for wireless communication and radar communication, and spin transfer torque RAM (STT-RAM) for data/information storage [6, 7].

The STO is a nanosized spintronic device capable of microwave generation at frequencies in the 1-65 GHz range with high quality factors [8-13]. It is compatible with the back-end flow of a standard Si process, and suitable for integration into high-frequency CMOS, thanks to its simple and compatible material stack. Although the STO is very promising for future telecommunication, its very weak power has to be dramatically improved before it can truly find practical use as a radio-frequency device. One possibility is the synchronization of two or more STOs to both increase the microwave power and further increase the quality factor. Synchronization of pairs of separate noncontact STOs in parallel that share a common magnetic layer can phase-lock with each other leading to a coherent summation of the individual amplitudes of each STO signal [14, 15]. Synchronization of serially connected STOs has also been theoretically suggested [16]. In this configuration, synchronization relies on phase locking between the STOs and their self-generated alternating current [8-13, 16]. However, very limited simulations have been performed on the synchronization of serially connected STOs network, due to the grand challenge of the physical problem and the requirement of extremely heavy computing power [8, 16].

In this paper, we have developed a parallel macrospin simulation code to study an array of phase coherent STOs. The parallel code is applied to study the effect of circuit phase shift on the phase-locking of serially connected STOs. It is shown that the STO network can be completely synchronized by tuning the circuit  $I$ - $V$

---

<sup>+</sup> Corresponding author.  
E-mail address: yanzhou@hkit.edu.hk.

phase. The corresponding synchronization phase diagram as a function of applied dc current and the Gaussian distribution of the STOs' intrinsic frequencies are also illustrated. The present work provides the first detailed study of synchronization of a large assembly of STOs, which we believe is essential for utilizing STO-based devices for telecommunication application.

## 2. Implementation of Parallel Macrospin Simulation code

### 2.1 Macrospin Model and description

We consider  $N$  electrically coupled STOs (to be called  $\text{STO}^{(i)}$  hereinafter) connected in series and coupled to a dc current  $I_t$  and to a resistive load  $R_C$  (see Fig. 1). Each of the STOs is a typical trilayer system consisting of a thick magnetically fixed layer, nonmagnetic spacer layer, and a thin free (sensing) layer (for example, Co/Cu/Co nanopillar in the experiment by Kiselev et al. [17]). The instantaneous current flowing through the STO branch  $I$  is given by the circuit condition [8]

$$I = \frac{R_C I_t}{R_C + N(R_P + R_{AP})/2 - (R_{AP} - R_P)/2 \sum_{i=1}^N \cos \theta_i(t - \tau)} \quad (1)$$

where  $R_P$  and  $R_{AP}$  are the resistances of the STO in its parallel and antiparallel magnetic configuration respectively.  $\theta_i(t) = \cos^{-1}(\hat{m}_i \cdot \hat{M}_i)$  is the angle between the magnetization of the fixed layer  $\hat{M}_i$  and that of the free layer  $\hat{m}_i$  of the  $i$ -th STO. In a real experimental setup, there can be a phase shift (or time delay) between the magnetization induced voltage change and the resulting current variations. Therefore we introduce a time delay parameter  $\tau$  in (1) to model the realistic situations, which may physically correspond to the incorporation of a phase shifter in Fig. 1 to artificially modulate the I-V phase shift [8].

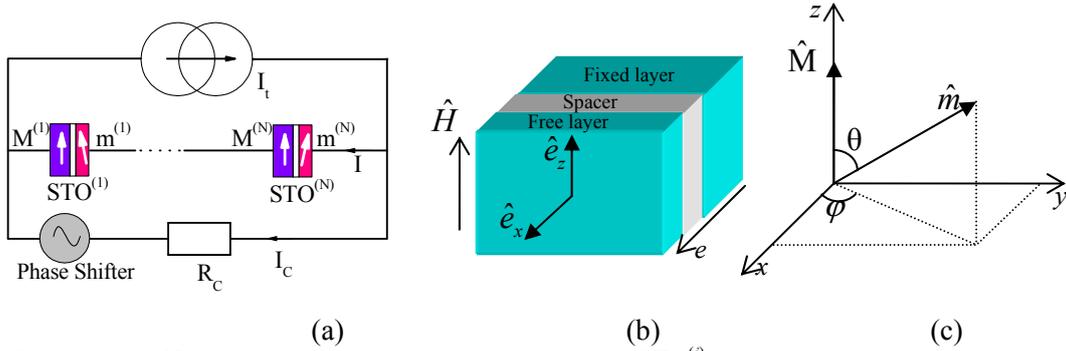


Fig. 1 (a) Sketch of the  $N$  serially connected oscillators (labelled by  $\text{STO}^{(i)}$  where  $i$  is the  $i$ th oscillator in the network) and coupled to a load  $R_C$  (throughout this letter,  $R_C=20 \Omega$ ). (b) Schematic structure of the oscillator with a nonmagnetic spacer layer sandwiched between a “fixed” ferromagnetic layer  $\mathbf{M}$  and a “free” ferromagnetic layer  $\mathbf{m}$ . The unit magnetization vector of the fixed layer and free layer are represented by  $\mathbf{M}$  ( $\mathbf{M}||Z$ ) and  $\mathbf{m}$ , respectively. The  $y$ - $z$  plane is the easy plane for  $\mathbf{m}$  and  $z$  axis is the easy axis. The current  $I$  is defined as positive when the negative electrons flow from the fixed layer to the free layer.

Magnetization dynamics of the free layers of the STO can be described by the generalized Landau-Lifshitz-Gilbert–Slonczewski (LLGS) equation [8-13, 16]

$$\frac{d\hat{m}_i}{dt} = -|\gamma|\hat{m}_i \times H_{eff}^{(i)} + \alpha\hat{m}_i \times \frac{d\hat{m}_i}{dt} + T_{STT} \quad (2)$$

where  $\gamma$  the gyromagnetic ratio,  $\alpha$  the damping coefficient,  $\mu_0$  the magnetic vacuum permeability,  $\eta$  the spin transfer efficiency,  $M_S$  the free layer saturation magnetization, and  $V$  stands for the volume of the free layer.  $H_{eff}^{(i)} = H_{app}\hat{e}_x + H_k(\hat{m}_i \cdot \hat{e}_x) - 4\pi M_S(\hat{m}_i \cdot \hat{e}_z)\hat{e}_z$  is the effective magnetic field acting on the free layer, which includes the applied in-plane magnetic field  $H_{app}$ , the uniaxial magnetic anisotropy field  $H_k$ , and the out-of-plane demagnetization field.  $\hat{e}_x$  and  $\hat{e}_z$  are the unit vectors along  $x$  (in-plane easy-axis) and  $z$  (out-of-plane), respectively. The last term in (2) is the Slonczewski spin torque term

$$T_{STT}^{(i)} = \frac{|\gamma|\hbar\eta I}{2\mu_0 M_S eV} \hat{m}_i \times (\hat{m}_i \times \hat{M}_i) \quad (3)$$

where  $I$  is the instantaneous current flowing through the STO branch, as given by (1). The instantaneous current  $I$  is also the only term of (2) that couples the different STOs.

We assume a Gaussian distribution in the anisotropy fields of the STOs due to the unavoidable process variations (especially due to the lithography limitation).

## 2.2 Implementation of parallel simulation code

The Parallel Computer Center (PDC) at KTH hosts the fastest supercomputer available in Sweden, and actively takes part in major international projects to develop high-performance computing and parallel computing. It primarily offers easily accessible computational resources that cater for the needs of all the postgraduate students and research staff of KTH. Our parallel code is implemented and benchmarked on the Dell Xeon cluster Lenngren of PDC. The basic structure of the main loop of the macrospin code is illustrated in Fig. 2, and will be detailed as below.

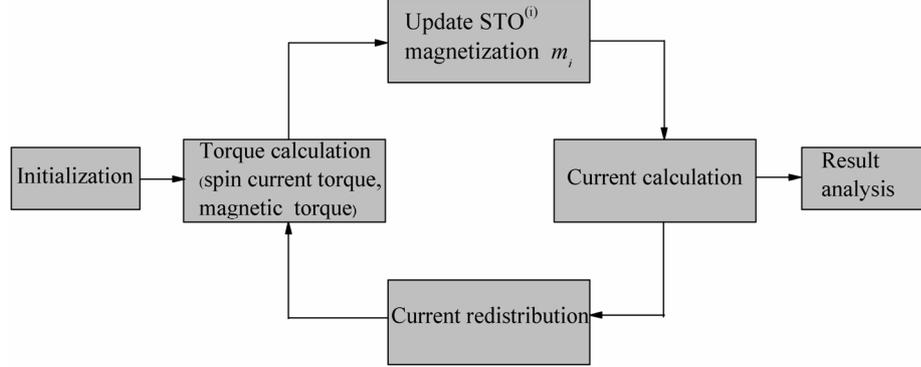


Fig. 2 Structure of the main loop of the macrospin simulation code including initialization, torque calculation, STO magnetization update, current calculation, current redistribution to slave nodes and final result analysis.

### Initialization

The free layer magnetization of STO<sup>(i)</sup>  $m_i(\theta_i, \phi_i)$ , anisotropy field  $H_k^{(i)}$ , demagnetization field  $H_d$ , and applied magnetic field  $H_{app}$  are initialized at  $t=0$ . Some material parameters of typical Co/Cu/Co nanopillar are adopted in our simulation [8-13, 16]:  $\alpha=0.007$ ,  $\gamma=1.85 \times 10^{11}$  Hz/T,  $M_S=1.27 \times 10^8$  A/m,  $\eta=0.35$ ,  $H_{app}=0.2$  T,  $H_d=1.6$  T,  $R_P=10 \Omega$ ,  $R_{AP}=11 \Omega$ . In the simulation codes, the anisotropy field of the free layer of STO<sup>(i)</sup>  $H_k^{(i)}$  follow the Gaussian distribution with mean  $\mu=0.05$  T and standard deviation  $\sigma$  is swept from 0 to 0.17 T ( $\sim \mu/3$ ). In this step, the  $N$  oscillators are divided into  $P$  groups, where  $P$  is the number of computational processors/nodes. The code is implemented in the way of providing the most equitable distribution of oscillators among the processes, and the computational load allocation is performed by the master node.

### Torque calculation

The first term on the right-hand side of (2) is a conventional magnetic torque. The ‘‘Gilbert damping’’ term  $\alpha \hat{m}_i \times \frac{d\hat{m}_i}{dt}$  in (2) includes the energy dissipation mechanisms, such as coupling to lattice vibrations and spin-flip scattering. Spin-transfer torque  $T_{STT}$  accounts for non-equilibrium processes that cannot be described by energy functional. It can counteract the damping term, and drive the STO into a steady precessional state [3]. For each slave node, it only calculates the torques of the corresponding STOs, which are assigned by the master node in the *initialization* step.

### Update magnetization $\hat{m}_i$

At  $t=t_0$ , the rate of magnetization variation  $\frac{d\hat{m}_i}{dt}$  is calculated in the step ‘‘Torque calculation’’ by (2). The STO magnetization at the next simulation step  $\hat{m}_i(t_0 + dt)$  can then be solved by employing a fourth-order Runge-Kutta algorithm with a calculation step of 0.1 ps. We have numerically checked that the results do not depend on the time step  $dt$  when  $dt < 0.5$  ps. It should be noted that each node is only responsible for updating the magnetizations of the STOs which are allocated by the master node at *initialization*.

### Current calculation

The current flowing through STOs branch  $I$  consists of both dc and ac components, arising from the current sharing between the STO branch and the load.

In this step, the master node needs to “gather” the individual magnetization polar angle  $\theta_i(t_0+dt)$  calculated from all the computational nodes in the previous step and then sum them up to calculate the total current flowing through the STO branch  $I(t_0+dt)$  by (1).

#### Current redistribution

Then the total current value is “broadcast” to all the nodes in order to proceed to the consequent step in the loop of Fig. 2 - “Torque calculation”. This is implemented by the “Allreduce” MPI function.

#### Result analysis

The aforementioned four iteration steps would be stopped if the convergence criterion of the simulation is met (i.e. the system reaches its steady state).

All the slave nodes need to send the calculation data, e.g., the precession frequency of each STO, to the master node. The master node would receive all the data collected from the slave nodes and then check whether all the STOs are synchronized. This is implemented by the “Reduce” MPI function.

### 2.3 Benchmark results of parallel simulation code

We use the MPI message passing library to realize the message-passing parallel programming model. In this model, we use a standard gnu Fortran compiler to generate an executable file to link the MPI message passing library. In the parallel-benchmarks, we measure the speedup and efficiency of the parallel code. The speedup  $S$  and efficiency  $E$  are defined as following:

$$S = T_1/T_p, \quad E = T_1/(PT_p), \quad (4)$$

where  $T_1$  is the total simulation time of the sequential code and  $T_p$  is the total running time of the parallel code for  $P$  processors, respectively.

Table 1 The effect of computational nodes on parallel speedup and efficiency.

No. of Nodes	Computation time (h)	Speedup	Efficiency
1	1133.3	0.98	0.98
2	575	1.95	0.98
4	291.7	3.79	0.95
8	158.3	7.10	0.89
16	91.7	11.96	0.75

Table I shows the influence of computational nodes on parallel speedup and efficiency. The number of STOs is as many as 1600. It can be seen that the parallel efficiency decreases with increasing the computational nodes, i.e.,  $E$  decrease from 0.98 for two nodes to 0.75 for 16 nodes. This is due to the increasing portion of the communication time of the processors (computation/communication ratio) when  $P$  is increased.

Table 2 The Influence Of Computational Load On Parallel Speedup And Efficiency.

No. of STOs	Computation time (h)	Speedup	Efficiency
100	27.5	2.78	0.69
200	44.2	3.25	0.81
400	80.8	3.51	0.88
800	150	3.73	0.93
1600	291.7	3.80	0.95

Table II shows the influence of computational load on parallel speedup and efficiency. The number of computational nodes is a constant ( $P=4$ ) whereas the total number of STOs is varied from  $N=100$  to  $N=1600$ . It can be seen that parallel efficiency  $E$  increases from 0.69 for  $N=100$  to 0.95 for  $N=1600$ . Thereby it is concluded that the parallel computing code is more suitable for studying the nonlinear dynamics and phase-locking of a large assembly of STOs.

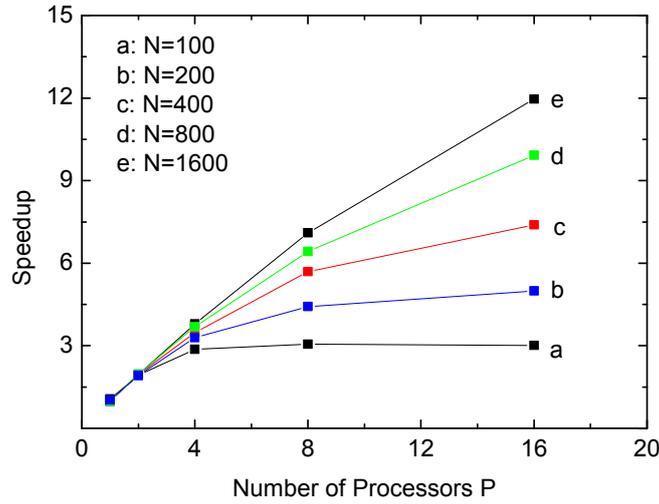


Fig. 3 Parallel speedup for different number of STOs (N) and computational nodes (P).

Fig. 3 shows the speedup versus the number of the computational nodes with varying the problem size *i.e.* the number of STOs. For a constant number of computational nodes, parallel speedup will be improved by increasing the computational load, *i.e.*,  $S(e) \geq S(d) \geq S(c) \geq S(b) \geq S(a)$  regardless of  $P$ . In general, the parallel speedup will increase monotonically with the number of nodes for a given computational load (except for  $N=100$ ). However, parallel effectiveness (the slope of the curve) decreases when the number of nodes increases. For  $N=100$ , parallel speedup  $S$  first increases with  $P$  and then reaches its maximum value at  $P=8$ . For  $P > 8$ , it slows down rather than speedup. From the definition of parallel speedup  $S$ , we can understand the dependencies of  $S$  on computational load and nodes. The parallel computational time for  $P$  processors can be expressed as follows:

$$T_P = T_1/P + T_0 \quad (5)$$

where  $T_0$  include the communication time between different nodes, the extra time due to the parallelization of codes and the extra time due to the load imbalance *etc.* Apparently  $T_0$  will increase with  $P$ . Therefore,

$$S = P \left( \frac{1}{1 + \frac{PT_0}{T_1}} \right) = \frac{1}{\frac{1}{P} + \frac{T_0}{T_1}} \quad (6)$$

From (6), we can see that for a given problem where  $T_1$  is a constant,  $S$  doesn't increase linearly with  $P$ . Considering that  $T_0$  will increase with  $P$ , it can be inferred that the  $S$  will increase with  $P$  for small  $P$ , but eventually decrease with  $P$  after reaching its maximum value. On the other hand, increasing the computational load (Consequently,  $T_1$  increases) will always give a larger  $S$ .

### 3. Applications: Synchronization of STOs Array

The physical problem of interest of the code is the synchronization of serially connected STOs array as shown in Fig. 1a. To the best of our knowledge, such simulations have only been attempted for very limited number of STOs due to the need of extremely demanding computing power [8, 12, 16]. In the past, we have successfully demonstrated that the synchronized state of STO pairs develops the highest robustness by tuning the total circuit  $I$ - $V$  phase shift [8, 12]. In this work, similar simulations have been carried out but for a much larger number of STOs. In Fig. 4, the synchronization of 100 STOs is shown as a function of applied current and standard deviation of the anisotropy fields of the STOs for (a)  $\tau=0$  and (b)  $\tau=0.05$  ns. At  $\tau=0$ , the system can only be completely phase-locked (the white region) for extremely small intrinsic frequency dispersion (or anisotropy field distribution) at the current range of about 0.3-9.4 A. From a manufacturing point of view, Fig. 4a is quite disappointing since this amount to extremely high demands of pattern fidelity on the lithography process. By introducing a current-voltage phase shift of about  $120^\circ$  which corresponds to  $\tau=0.05$  ns in the circuit, the intrinsically very weak locking mechanism can be dramatically improved (Fig. 4b). The white synchronization region dominates over the entire phase diagram for  $\tau=0.05$  (Fig. 4b) in contrast to the very

limited synchronization region at zero  $I$ - $V$  phase ( $\tau=0$  in Fig. 4a), hence relaxing the severe demands on device variability. We got similar simulation results for 1600 serially connected STOs, which to date is the most STOs ever simulated.

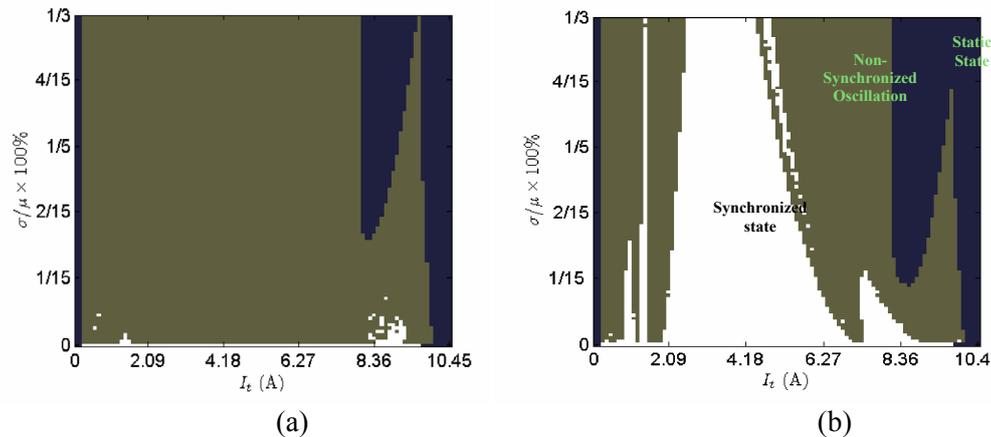


Fig. 4 Simulated synchronization phase diagram as a function of the total current  $I_t$  and  $\sigma$  to  $\mu$  ratio of the normal distribution of anisotropy fields (a)  $\tau=0$  ns; (b)  $\tau=0.05$  ns. Black: at least one STO is in static state; gray: the STOs show asynchronous oscillations; white: all the STOs are in synchronized oscillating state, i.e., oscillation frequency for these 100 STOs are the same.

## 4. Conclusions

We have developed the parallel simulation package for studying the synchronization robustness of current-mediated STOs in serial connection. We have benchmarked the performance and evaluated the parallel efficiency of our home-made parallel codes. For a given problem size ( $N$  is fixed), the parallel efficiency will decrease with increasing computational nodes. For a given number of computational nodes ( $P$  is fixed), the parallel efficiency will increase with the problem size, indicating the appropriateness of employing such large-scale parallel computing for studying the synchronization of a large assembly of STOs.

## 5. Acknowledgment

Y. Z. thanks the technical support from Parallel Computer Center (PDC) of KTH, Sweden.

## 6. References

- [1] X. Guo, E. Ipek, and T. Soyata, "Resistive Computation: Avoiding the Power Wall with Low-Leakage, STT-MRAM Based Computing," Intl. Symp. on Computer Architecture (ISCA'10), Proceedings of the 37th annual international symposium on Computer architecture, Jun. 2010, pp. 371-382, doi: doi.acm.org/10.1145/1815961.1816012.
- [2] B. Dieny et al., "Spin-transfer effect and its use in spintronic components," International Journal of Nanotechnology, vol. 7, 2010, pp. 591 - 614, doi: 0.1504/IJNT.2010.031735.
- [3] J. Z. Sun, "Spin angular momentum transfer in current-perpendicular nanomagnetic junctions", IBM Journal of Research and Development, vol. 50, Jan. 2006, pp. 81–100.
- [4] A. Fert, "Nobel Lecture: Origin, development, and future of spintronics", Reviews of Modern Physics, vol. 80, Dec. 2008, pp. 1517–1530, doi: 10.1103/RevModPhys.80.1517.
- [5] I. Zutíć and J. Fabian, "Spintronics: Silicon twists", Nature, vol. 447, May 2007, pp. 268-269, doi: 10.1038/447269a.
- [6] S. H. Kang, "Recent advances in spintronics for emerging memory devices", Journal of the minerals, Metals and Materials Society, vol. 60, Sep. 2008, pp. 28-33, doi: 10.1007/s11837-008-0113-0.
- [7] George Bourianoff, Joe E. Brewer, Ralph Cavin, James A. Hutchby, Victor Zhirnov, "Boolean Logic and Alternative Information-Processing Devices," Computer, vol. 41, pp. 38-46, May 2008, doi:10.1109/MC.2008.145

- [8] J. Persson, Y. Zhou, and J. Akerman, "Phase-locked spin torque oscillators: Impact of device variability and time delay", *Journal of Applied Physics*, vol. 101, pp. 09A503, Apr. 2007, doi: 10.1063/1.2670045.
- [9] Y. Zhou, J. Persson, S. Bonetti, and J. Akerman, "Tunable intrinsic phase of a spin torque oscillator", *Applied Physics Letters*, vol. 92, pp. 092505, Mar. 2008, doi: 10.1063/1.2891058.
- [10] Y. Zhou, J. Persson, and J. Akerman, "Intrinsic phase shift between a spin torque oscillator and an alternating current", *Journal of Applied Physics*, vol. 101, pp. 09A510, May 2007, doi: 10.1063/1.2710740.
- [11] Y. Zhou, S. Bonetti, C. L. Zha, and J. Akerman, "Zero-field precession and hysteretic threshold currents in a spin torque nano device with tilted polarizer", *New Journal of Physics*, vol. 11, pp. 103028, Jul. 2009, doi: 10.1088/1367-2630/11/10/103028.
- [12] Y. Zhou, S. Bonetti, J. Persson, and J. Akerman, "Capacitance Enhanced Synchronization of Pairs of Spin-Transfer Oscillators", *IEEE Transactions on Magnetics*, vol. 45, pp. 2421-2423, Jun. 2009, doi: 10.1109/TMAG.2009.2018595.
- [13] Y. Zhou, F. G. Shin, B. Guan, and J. Akerman, "Capacitance Effect on Microwave Power Spectra of Spin-Torque Oscillator With Thermal Noise", *IEEE Transactions on Magnetics*, vol. 45, pp. 2773-2776, Jun. 2009, doi: 10.1109/TMAG.2009.2020556.
- [14] F. B. Mancoff, N. D. Rizzo, B. N. Engel, S. Tehrani, "Phase-locking in double-point-contact spin-transfer devices", *Nature*, vol. 437, Sep. 2005, pp. 393-395, doi: 10.1038/nature04036.
- [15] S. Kaka, M. R. Pufall, W. H. Rippard, T. J. Silva, S. E. Russek, J. A. Katine, "Mutual phase-locking of microwave spin torque nano-oscillators", *Nature*, vol. 437, Sep. 2005, pp. 389-392, doi: 10.1038/nature04035.
- [16] J. Grollier, V. Cros, and A. Fert, "Synchronization of spin-transfer oscillators driven by stimulated microwave currents", *Physical Review B*, vol. 73, pp. 060409, Feb. 2009, doi: 10.1103/PhysRevB.73.060409.
- [17] S. I. Kiselev et. al., "Microwave oscillations of a nanomagnet driven by a spin-polarized current", *Nature*, vol. 425, pp. 380-383, Sep. 2003, doi: 10.1038/nature01967.