

A Method for Time Error Correction in SIL Simulation

Kyungnam Kim¹⁾ Seungkyu Yoo¹⁾ Hyunrak Ko¹⁾ Sumin Park¹⁾

1) 1)Hyundai-Autoever

12, Teheran-ro 113-gil, Gangnam-gu, Seoul, Republic of Korea (E-mail: kyungnam.kim@hyundai-autoever.com)

KEY WORDS: software and its underlying technologies, HILS, SILS , virtualization, HIL-SIL Coupling[E3]

In Virtual ECU-based verification environments, Hardware-in-the-Loop (HIL) and Software-in-the-Loop (SIL) systems are often coupled together. HIL operates on a fixed cycle (e.g., 10 ms) under a Real-Time OS, while SIL operates on Simulation Time within a general-purpose OS (e.g., C++), making precise time synchronization between the two essential. Conventionally, the Sleep() function has been used to implement the SIL waiting period. However, since Sleep() delegates timing to the OS scheduler, it suffers from scheduler priority adjustments and context-switching delays: for wait times below ~1 ms, the discrepancy between actual and requested time increases sharply, degrading HIL-SIL synchronization reliability. As an alternative, Spin Lock achieves ~100 ns accuracy via a repeated while-loop comparison but occupies the CPU continuously, causing severe CPU efficiency loss. Thus, Sleep() offers high CPU efficiency but low accuracy, while Spin Lock offers high accuracy but low CPU efficiency — an inherent trade-off.

This paper proposes a Precision Timer that combines both advantages. It repeatedly calls the low-cost Sleep() while estimating its statistical execution error (mean and sample standard deviation) in real time using Welford's online algorithm, a low-cost, single-pass statistical method. Once the remaining time narrows to within the estimated error range, the method switches to Spin Lock to achieve microsecond (μ s)-level precision — minimizing HIL-SIL time error while preserving CPU efficiency throughout most of the wait.

The mean m_n and sample variance s_n^2 of the Sleep() error up to the n -th trial are updated as:

$$\Delta_n = x_n - m_{n-1}, m_n = m_{n-1} + \frac{\Delta_n}{n}, M_n = M_{n-1} + \Delta_n(x_n - m_n), s_n^2 = \frac{M_n}{n-1}$$

The estimated error is $e_n = m_n + \sqrt{s_n^2}$. The method switches from Sleep() (Timer A) to Spin Lock (Timer B) once the remaining time $T \leq e_n$: (1) initialize $e, m, s^2, n = 1$; (2) compare T with e ; (3) if $T > e$, execute Sleep(1 ms) and update statistics, then repeat; (4) if $T \leq e$, precisely wait via Spin Lock until the target time is reached.

Setting the Sleep unit interval t is important: too large increases post-switch Spin Lock cost; too small reduces statistical sample reliability. In conclusion, the proposed Precision Timer eliminates the time error inherent to Sleep() alone while keeping CPU utilization equivalent to Sleep()-only usage, achieving microsecond-level HIL-SIL synchronization. Future work includes dynamic interval adjustment, refined error-distribution modeling, system-load-based calibration, and multi-controller error-sharing to further improve reliability across diverse HIL-SIL environments..

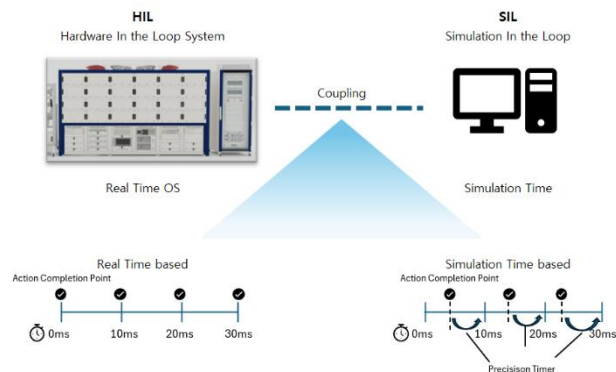


Fig.1 Representative example