# Supplement of “Physical-State-Aware Dynamic Slack Management for Mixed-Criticality Systems”

Hoon Sung Chwa[1], Kang G. Shin[1], Hyeongboo Baek[2], and Jinkyu Lee[2]

[1]Electrical Engineering and Computer Science, The University of Michigan, Ann Arbor, Michigan, U.S.A.

[2]Department of Computer Science and Engineering, Sungkyunkwan University (SKKU), Republic of Korea.

## Appendix

### A. *Example of illustrating the usage of slacks*

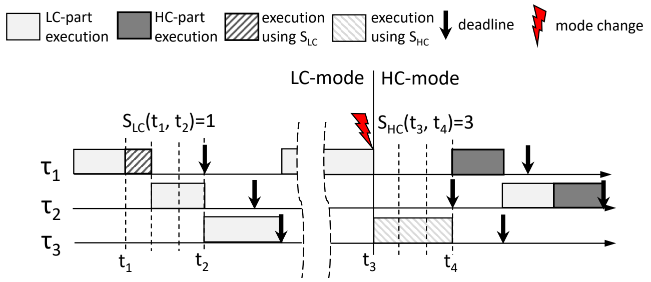

Fig. 3. An example of illustrating the usage of $LC$- and $HC$-mode slacks.

Fig. 3 illustrates how $LC$- and $HC$-mode slacks can be utilized. In the example, a job of $HC$ task $\tau_1$ completes its $LC$-part execution at $t_1$, and $LC$-mode slack in $[t_1,\ t_2)$ is 1. Then, according to Lemma 2, the job can execute its $HC$-part execution by using $LC$-mode slack without triggering a mode-switch while satisfying other jobs' $LC$-part execution before their deadlines. After a mode-switch at $t_3$, $HC$-mode slack in $[t_3, t_4)$ is 3. Then, according to Lemma 3, a job of $LC$ task $\tau_3$ can execute its $LC$-part execution by using $HC$.

### B. *Example of slack calculation in Algorithm 2*

Figs. 4(a)–(c) illustrate how to calculate $LC$-mode slack $S^*_{LC}(0,8)$ in [0,8). In Fig. 4(a), at time 0, we first plan to defer $\tau_3$'s execution until after $d_1$ but by its deadline $d_3$. Likewise, in Fig. 4(b), we try to fit $\tau_2$'s execution between $d_1$ and $d_2$ while guaranteeing that the sum of utilization in the interval is less than or equal to one. Some portion of $\tau_2$'s execution does not fit and must execute before $d_1$, requiring use of interval [6,8). In Fig. 4(c), after assigning the remaining portion of $\tau_1$'s execution, we can calculate $S^*_{LC}(0,8)$ as the amount of idle time in [0,8), which is 2.

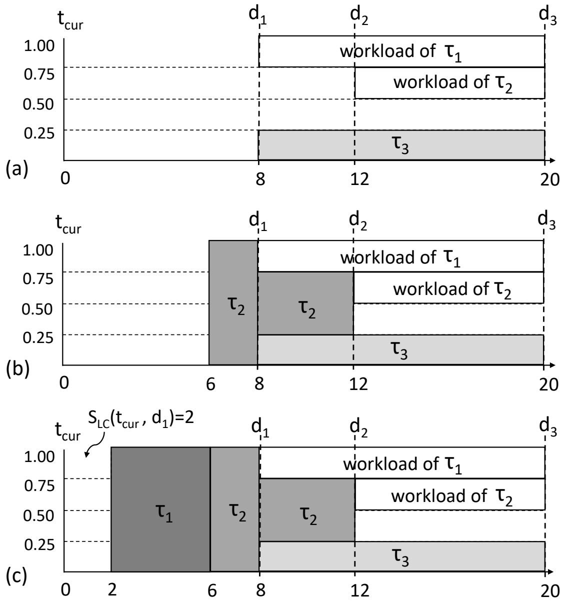

Fig. 4. An example of slack calculation $S_{LC}(0, 8)$.