EVENT-TRIGGERED SLIDING MODE OBSERVER

B. Sensor Side Implementation

Sliding mode observer:

$$\begin{aligned}\dot{\hat{z}}_1(t) &= \widetilde{A}_{11}\hat{z}_1(t) + \widetilde{A}_{12}\hat{z}_2(t) + L_1\widetilde{C}_{11}\tilde{z}_1(t) + \widetilde{A}_{12}\tilde{z}_2(t)\\ \dot{\hat{z}}_2(t) &= \widetilde{A}_{21}\hat{z}_1(t) + \widetilde{A}_{22}z_2(t) + u(t_\ell^z) + K\text{sign}\left(\tilde{z}_2(t)\right).\end{aligned} \tag{13}$$

Theorem 3: Consider the plant (3) and the observer (13). Let $c_1 > 0$ and $c_2 > 0$ be any scalars, and $\alpha_s > 0$ be some given constant. Suppose that the pair $(\widetilde{A}_{11}, \widetilde{C}_{11})$ is observable. Moreover, assume that $z(0), \hat{z}(0) \in \Omega(c_1, c_2)$ and $z(t)$ is bounded for all $t \geqslant 0$. Then, there exist $K > 0$ and $\tau_s > 0$ such that

1) the trajectory $\tilde{z}(t)$ is bounded for all $t \geqslant 0$, and moreover $\tilde{z}(t) \to 0$ as $t \to +\infty$;
2) $t_{\ell+1}^z - t_\ell^z \geqslant \tau_s$ for all $\ell \in \mathbb{Z}_{\geqslant 0}$.

$$t_{\ell+1}^z = \inf\left\{t > t_\ell^z : \|\hat{z}(t_\ell^z) - \hat{z}(t)\| \geqslant \sigma_s\alpha_s\right\}, \quad \ell \in \mathbb{Z}_{\geqslant 0}$$

$$t_0^z = 0$$

The triggering parameter has no role in the estimation accuracy, the selection of this parameter does not depend on the observer dynamics, albeit it is an integral part of it.

CONCLUSION

In this article, we proposed an event-based sliding mode observer for estimating the states practically, i.e., with any arbitrary accuracy. It was possible by the proper tuning of event parameters, possibly online. We discussed the implementation of the observer in two different architectures. In the first one, we proved that the states are estimated practically irrespective of external disturbances. However, in sensor-side implementation, the estimation error converges to zero as the outputs are continuously fed to the observer. The proposed sliding mode observer can be useful for various applications, e.g., fault detection and isolation, output feedback stabilization, detection of cyber-attacks, etc.

Problem：

Online tuning of event parameters?