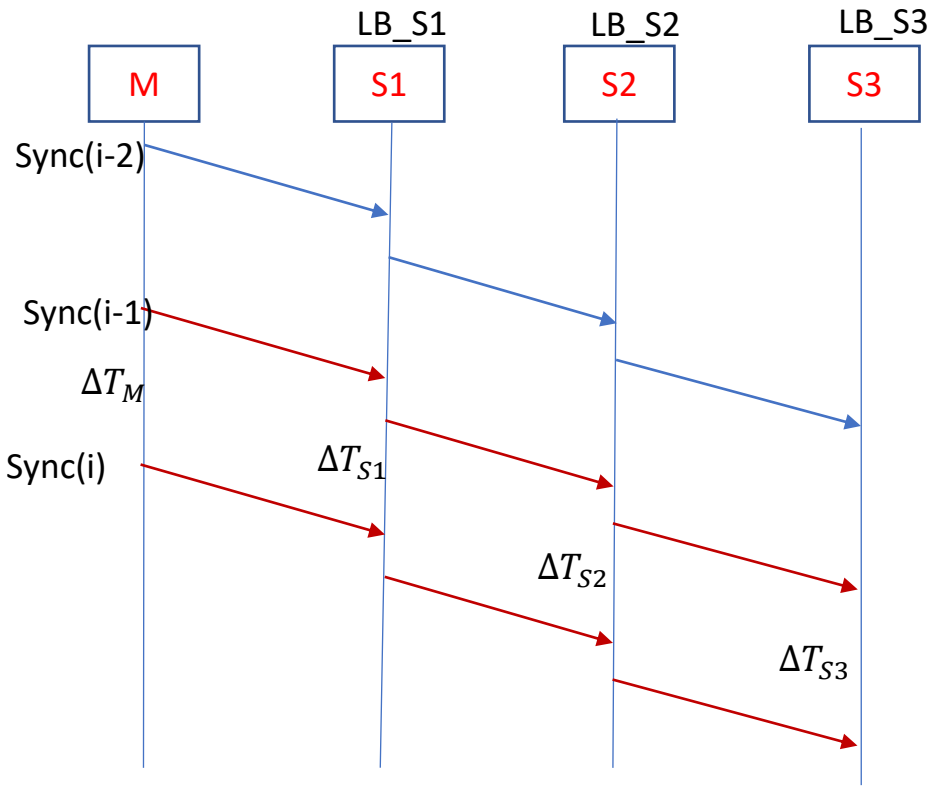
Error Propagation in RR Calculation using Sync Messages

Dragan Obradovic, Siemens AG

RR calculation via Sync messages



$$Definition:\ RR_{M,Sn}^{i-1,i} = \frac{\widehat{M}_{n-1}^{i} - \widehat{M}_{n-1}^{i-1}}{S_n^i - S_n^{i-1}}$$

$\widehat{M}_{n-1}^{i}$: estimated Master time passed on by element „n-1“ in the Sync message number „i“

M^i: the Master Time passed on by the GM in the Sync Message “i”

S_n^i: the local time of Slave n at the arrival of Sync message i

LB: sum of estimated pDelay and Residence time in LocalClock frame

$$RR_{M,S1}^{i-1,i} = \frac{M^i - M^{i-1}}{S_1^i - S_1^{i-1}} = \frac{\Delta T_M^{i-1,i}}{\Delta T_{S1}^{i-1,i}}$$

$$RR_{M,S2}^{i-1,i} = \frac{\widehat{M}_1^i - \widehat{M}_1^{i-1}}{S_2^i - S_2^{i-1}} = \frac{(M^i + RR_{M,S1}^{i-1,i} \cdot LB_{S1}^i) - (M^{i-1} + RR_{M,S1}^{i-2,i-1} \cdot LB_{S1}^{i-1})}{\Delta T_{S2}^{i-1,i}}$$

$$RR_{M,S3}^{i-1,i} = \frac{\widehat{M}_2^i - \widehat{M}_2^{i-1}}{S_3^i - S_3^{i-1}} = \frac{(M^i + RR_{M,S1}^{i-1,i} \cdot LB_{S1}^i + RR_{M,S2}^{i-1,i} \cdot LB_{S2}^i) - (M^{i-1} + RR_{M,S1}^{i-2,i-1} \cdot LB_{S1}^{i-1} + RR_{M,S2}^{i-2,i-1} \cdot LB_{S2}^{i-1})}{\Delta T_{S3}^{i-1,i}}$$

RR calculation via Sync messages

$$RR_{M,Sn}^{i-1,i} = \frac{\widehat{M}_{n-1}^{i} - \widehat{M}_{n-1}^{i-1}}{S_n^i - S_n^{i-1}}$$

From the last page, we can generalize to slave “n”:

$$RR_{M,S_n}^{i-1,i} = \frac{\widehat{M}_{n-1}^{i} - \widehat{M}_{n-1}^{i-1}}{S_n^i - S_n^{i-1}} = \frac{\Delta T_M^{i-1,i}}{\Delta T_{Sn}^{i-1,i}} + \frac{\sum_{k=1}^{n-1}\left(RR_{M,Sk}^{i-1,i} \cdot LB_{Sk}^{i} - RR_{M,Sk}^{i-2,i-1} \cdot LB_{Sk}^{i-1}\right)}{\Delta T_{Sn}^{i-1,i}}$$

This is a recursive formula in both time and line-depth dimensions.

The terms in the sum are weighted differences (weights are LBs) over time (i.e. this is a discrete differentiation)!

The position of the element in the line determines the amount of the past information it carries along.

RR calculation via Sync messages – Recursion Transfer Function in z domain where RR calculating using consecutive Syncs

$$RR_{M,S_n}^{i-1,i} - RR_{M,S_n-1}^{i-1,i} = \left(\Delta T_M^{i-1,i} + \sum_{k=1}^{n-2} \left(RR_{M,Sk}^{i-1,i} \cdot LB_{Sk}^{i} - RR_{M,Sk}^{i-2,i-1} \cdot LB_{Sk}^{i-1} \right) \right) \cdot \left(\frac{1}{\Delta T_{Sn}^{i-1,i}} - \frac{1}{\Delta T_{Sn-1}^{i-1,i}} \right) + \frac{RR_{M,Sn-1}^{i-1,i} \cdot LB_{Sn-1}^{i} - RR_{M,Sn-1}^{i-2,i-1} \cdot LB_{Sn-1}^{i-1}}{\Delta T_{Sn}^{i-1,i}}$$

$$RR_{M,S_n}^{i-1,i} \approx RR_{M,Sn-1}^{i-1,i} \cdot \left(1 + \frac{LB_{Sn-1}^{i}}{\Delta T_{Sn}^{i-1,i}} \right) - RR_{M,Sn-1}^{i-2,i-1} \cdot \frac{LB_{Sn-1}^{i-1}}{\Delta T_{Sn}^{i-1,i}}$$

$$RR_{M,S_n}^{i-1,i}(z) \approx \frac{z \cdot \left(1 + \frac{LB_{Sn-1}^{i}}{\Delta T_{Sn}^{i-1,i}} \right) - \frac{LB_{Sn-1}^{i-1}}{\Delta T_{Sn}^{i-1,i}}}{z} \cdot RR_{M,Sn-1}^{i-1,i}$$

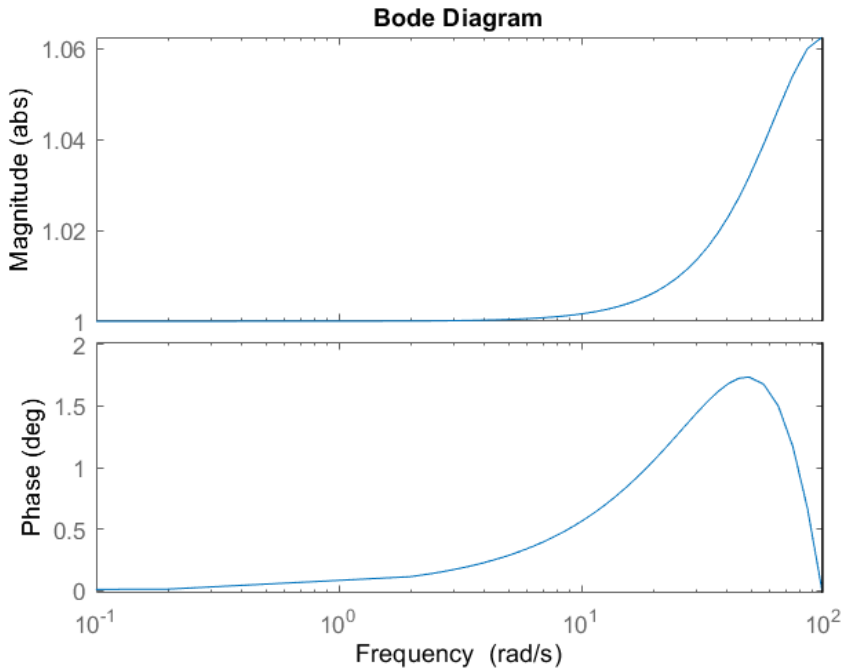
This recursive equation describes the additive “error” propagation over the nodes

Example: stamping error at GM s. t. $\Delta T_M^{i-1,i}$ - $\Delta T_{M-true}^{i-1,i}$ = err

RR calculation via Sync messages – Recursion Transfer Function in z domain where RR calculating using consecutive Syncs

$$RR_{M,S_n}^{i-1,i} \approx RR_{M,Sn-1}^{i-1,i} \cdot \left(1 + \frac{LB_{Sn-1}^{i}}{\Delta T_{Sn}^{i-1,i}}\right) - RR_{M,Sn-1}^{i-2,i-1} \cdot \frac{LB_{Sn-1}^{i-1}}{\Delta T_{Sn}^{i-1,i}}$$

$$RR_{M,S_n}^{i-1,i}(z) \approx \frac{z \cdot \left(1 + \frac{LB_{Sn-1}^{i}}{\Delta T_{Sn}^{i-1,i}}\right) - \frac{LB_{Sn-1}^{i-1}}{\Delta T_{Sn}^{i-1,i}}}{z} \cdot RR_{M,Sn-1}^{i-1,i}$$



Tsync=32ms, LB=1ms, RR with every Sync

MaxGain=1.0625

MaxGain^100= 429.4315

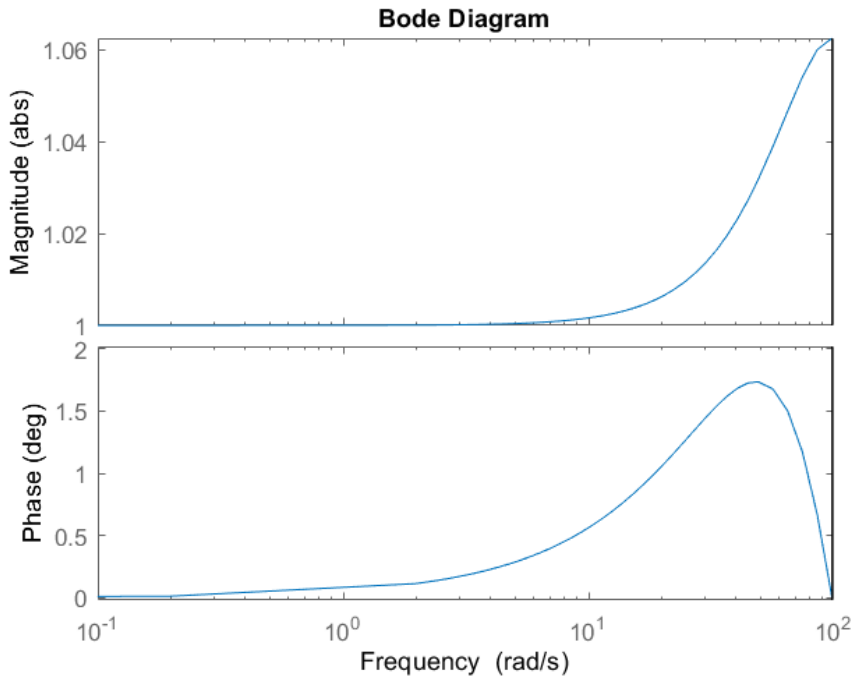
RR calculation via Sync messages – Recursion Transfer Function in z domain where RR calculating using consecutive Syncs

$$RR_{M,S_n}^{i-1,i} \approx RR_{M,Sn-1}^{i-1,i} \cdot \left(1 + \frac{LB_{Sn-1}^{i}}{\Delta T_{Sn}^{i-1,i}}\right) - RR_{M,Sn-1}^{i-2,i-1} \cdot \frac{LB_{Sn-1}^{i-1}}{\Delta T_{Sn}^{i-1,i}}$$

Tsync=32ms, LB=1ms, RR with every Sync

MaxGain=1.0625

MaxGain^100= 429.4315



Why this high-pass filter behavior?

→ Because there is a discrete differentiation

Low-pass filtering can help. This is exactly what we have done!

RR calculation via Sync messages – Recursion Transfer Function in z domain with RR calculation using every 7th Sync

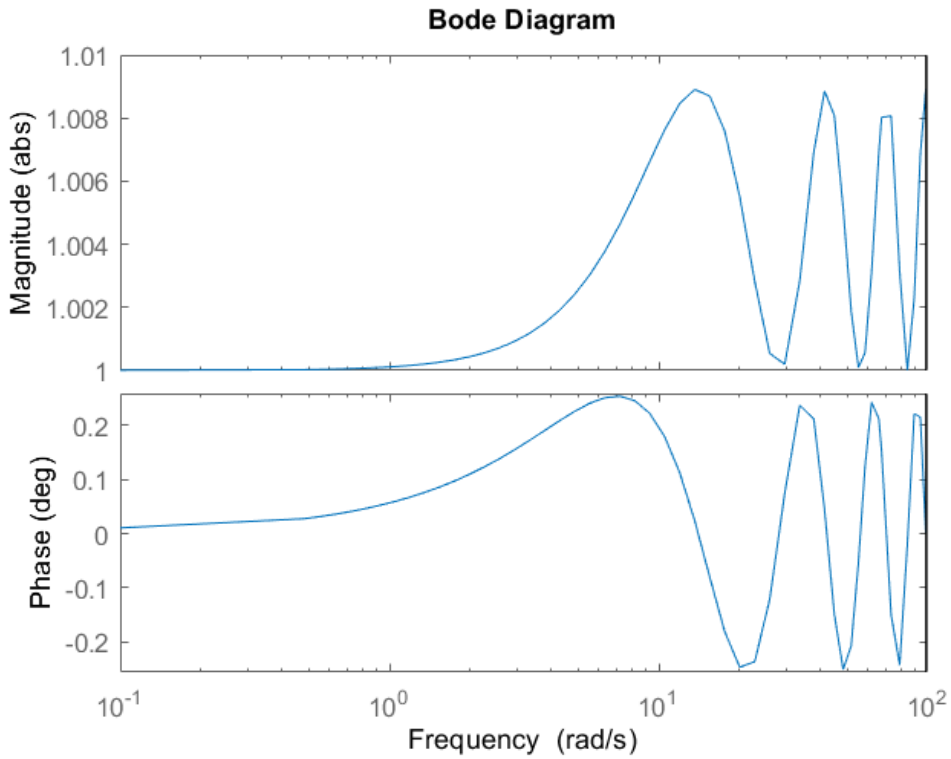
$$RR_{M,S_n}^{i-7,i} \approx RR_{M,Sn-1}^{i-7,i} \cdot \left(1 + \frac{LB_{Sn-1}^{i}}{\Delta T_{Sn}^{i-7,i}}\right) - RR_{M,Sn-1}^{i-14,i-7} \cdot \frac{LB_{Sn-1}^{i-7}}{\Delta T_{Sn}^{i-7,i}}$$

$$RR_{M,S_n}^{i-7,i}(z) \approx \frac{z\text{^}7 \cdot \left(1 + \frac{LB_{Sn-1}^{i}}{\Delta T_{Sn}^{i-7,i}}\right) - \frac{LB_{Sn-1}^{i-7}}{\Delta T_{Sn}^{i-7,i}}}{z\text{^}7} \cdot RR_{M,Sn-1}^{i-7,i}$$

Tsync=32ms, LB=1ms, **RR calculated with every 7th Sync**

MaxGain=1.0089

MaxGain^100= 2.4324 (instead of 429)

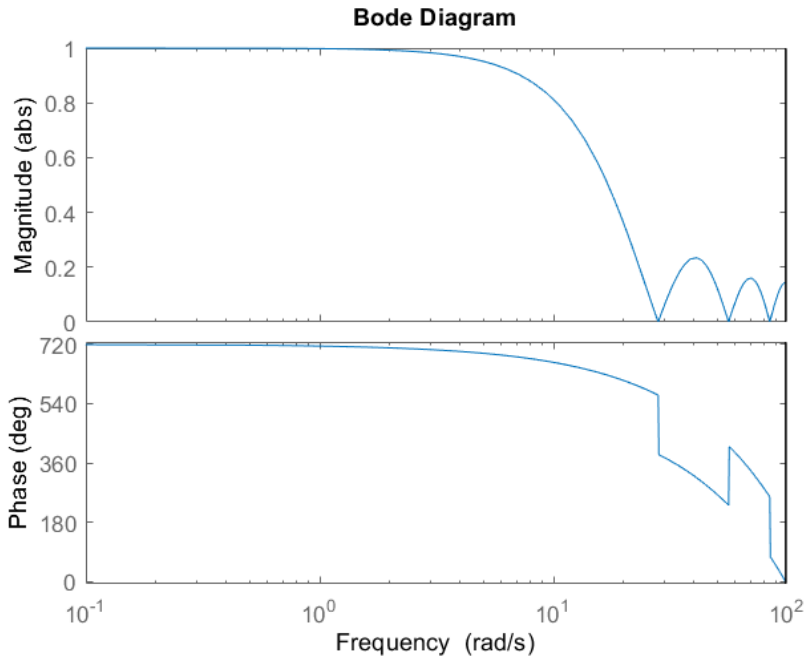


RR calculation via Sync messages – Recursion Transfer Function in z domain

with RR calculation using every 7th Sync *including filtering with a moving window of length M*

$$RR_{M,Sn}^{i-7,i}(z) \approx \frac{z\text{^}7 \cdot \left(1 + \frac{LB_{Sn-1}^{i}}{\Delta T_{Sn}^{i-7,i}}\right) - \frac{LB_{Sn-1}^{i-7}}{\Delta T_{Sn}^{i-7,i}}}{z\text{^}7} \cdot RR_{M,Sn-1}^{i-7,i}$$

Now implement a moving averaging filter of the length M=7 with a step of Tsync. The Bode plot of the filtered system is:



Tsync=32ms, LB=1ms, RR calculated with every 7th Sync, filtered

MaxGain=1

MaxGain^100= 1! → NO error accumulation

RR calculation using Sync Messages: Conclusion

- Calculation of RR at Sn via Sync messages {i-1,i} carries (depends on) the past information contained in the already calculated RRs with previous Sync messages
- The size of „carried past“ is determined by the position („n“) of the studied element in the line
- The discrete differentiation present in RR calculation is a high-pass filter whose frequency response depends on the ratio ResidenceTime/Tsync
- The mentioned high pass filter can amplify the high frequency noise which might lead to gain-peaking
- *Filtering (already applied) remedies this problem and there is no error accumulation*
- The RR calculation based on nRRs does not have “memory” in the sense of depending on the GM time carried in the previous messages, but has as a problem of using the old nRR information
- The proposed idea of letting Sync messages be used in calculating nRRs is promising.