CAN XL System Design – About Clock Tolerances and Edge Deviations

Dr. Arthur Mutter, Robert Bosch GmbH

When designing a CAN XL bus system, one main target is to achieve a reliable communication under all operating conditions. Therefore, the system designer needs to consider many constraints and to choose the proper bit timing configuration. The two most relevant constraints are the frequency tolerance *df* of the clock source and the asymmetry of the bit lengths caused by physical layer effects.

This paper derives the formulas to calculate the maximal clock tolerance *df* for CAN XL. Then it evaluates and compares the *df* of CAN FD and CAN XL. Furthermore, this paper adapts the metric called "phase margin", known from CAN FD, to CAN XL. This metric allows to assess the robustness of a CAN XL bus system, i.e., up to which extent physical layer effects can be allowed, without endangering the reliability of the CAN XL communication.

1. Introduction

A. Overview of CAN Evolution

For a reliable communication under all operating conditions on the bus the system designer must (i) choose a suitable bit timing configuration, (ii) check if the CAN clock source precision is sufficiently high (iii) be aware of the physical effects on the bus (e.g., bit asymmetry, ringing).

CAN CC [1] was first presented by Bosch in 1986 on an SAE Congress. This, at that time revolutionary communication protocol was the start of the CAN success story. The configuration of the CAN CC bit timing is explained in [2].

CAN FD [3] was first presented by Bosch in 2012. It allows to use much shorter bit times than with CAN CC, but also requires a more careful system design. The configuration of the CAN FD bit timing is explained in [4]. For CAN FD the authors in [5] list sources for a phase error, explain the clock tolerance formulas, and introduce a new metric called phase margin. [6] contains the formulas, recommendations, and a cookbook for CAN FD network design. [7] highlights the strength of the CAN FD error detection mechanism.

The specification of CAN XL started in December 2018 in the Special Interest Group CAN XL at CiA. CAN XL is being released as ISO Standard [12] in 2024. CAN XL supports new so called SIC XL transceivers [11] to achieve bit rates of up to 20 Mbit/s. [8] shows how to introduce CAN XL nodes into existing CAN networks. [9] and [10] highlight the numerous improved error detection mechanisms of CAN XL. CiA 612-1 [13] will be released in 2024 and gives valuable system design recommendations, as example for (i) clock frequency, tolerance, jitter, (ii) bit timing configuration, (iii) operation modes, etc.,

B. Paper Content

This paper consists of two parts.

In the first part we derive the conditions to calculate the accepted node clock frequency tolerance *df* for CAN XL. Then we evaluate the *df* with a set of bit timing configurations. Further we compare the results to CAN FD.

In the second part we derive the metric called "phase margin" [6] known from CAN FD also for CAN XL. This metric allows to assess the robustness of a CAN XL bus system, i.e., up to which extent physical layer effects are allowed.

C. Phase Error

CAN nodes synchronize on the incoming bit stream to enable the usage of low cost oscillators as CAN node clock source. The

tolerance range of the CAN node clock frequency depends on the used bit timing configuration.

The analogue effects on the physical layer (pins, transceiver, cabling, etc.) cause shifts of bit edges and unstable signal levels. Towards shorter bit times these effects have a larger impact on the robustness of the communication.

Phase Error definition: For each receiving node, the distance (time) between the detected position and the expected position of an edge is called the phase error of that edge [2].

Different phase error sources exist, which add up to the total phase error. This Subsection introduces these error sources, including their causes.

D. Error Classification

According [5] the error sources can be classified into two independent classes: accumulating and non-accumulating errors.

Accumulating Errors – Accumulating errors lead to a phase error that adds up over time, i.e., the phase error groves over time. The “Frequency Error” is the only accumulating error. It is caused by the frequency difference between the node clocks in sender and receiver. (Here we disregard the possibility for a second accumulating error, “wrong bit rate configuration”, where the user configured e.g., instead of 1 Mbit/s a bit rate of 1.05 Mbit/s.)

A CAN node (re)synchronizes at every falling edge (1➔0) on the bit stream of the transmitter to eliminate the accumulated phase error.

Non-Accumulating Errors – Non-accumulating errors are temporary shifts of bit edges or shifts of the internal view of a receiver. These errors occur sporadically and do not add up over time. Corresponding error sources are:

- Quantization Error – A CAN node samples an incoming bit stream once per time quantum. This leads to an error of at most one time quantum.
- Bit Symmetry Error – A bit is lengthened or shortened compared to how it was transmitted by the protocol controller, i.e. the received bits are asymmetric. This is caused for example by
  - CAN transceiver
  - network topology
  - electromagnetic disturbances
  - asymmetric rise and fall times of CAN RX and TX signals
  - jitter of node clock frequency (which effects receiving and transmitting nodes)

The phase error introduced by non-accumulating errors cannot be corrected via resynchronization.

2. Node Clock Frequency Tolerance (*df*) in CAN XL

A. Overview

The conditions to calculate the node clock frequency tolerance (*df*) for CAN XL are given in the ISO standard [12]. This Section derives these conditions and shows for each the corresponding worst case bit sequence. *df* depends solely on the bit timing configuration.

The worst-case bit sequences rarely occur. This means during normal operation (i.e., non-worst case) a node typically can allow a tolerance that is higher than the one calculated.

B. CAN XL Properties

The most important properties of CAN that are necessary to understand the derivation of the conditions for *df* are summarized next.

To synchronize on an incoming bit stream, a CAN node oversamples the received bits and corrects its internal view according to what it senses on the bus. A node synchronizes only on falling edges, which are edges from logical 1 to logical 0 signal level. As example, if a receiving CAN node recognizes a falling edge being earlier or later than expected, it corrects its internal view to be synchronous to the node transmitting the bits. [2] provides a detailed

description of the synchronization mechanisms in CAN CC. CAN FD and CAN XL use the identical mechanisms.

A CAN XL frame has two phases [8], like in CAN FD.

- **Arbitration Phase** – This phase contains the frame parts, where potentially several nodes may drive the bus simultaneously. In this phase, the restrictions regarding bus length and maximum bit rate are equal to those of CAN CC.
- **Data Phase** – In the data phase only a single transmitter exists. In this phase the CAN XL protocol has no restrictions regarding bus length or maximum bit rate. (This is also true for CAN FD.)

Each phase has its own bit timing configuration. The two bit timing configurations are independent of each other, except the constraint that the bit rate in the data phase has to be at least 2 times higher than in the arbitration phase.

CAN XL bit stuffing rules

- **Dynamic bit stuffing** – In the arbitration phase, from SOF bit to the beginning of the data phase, dynamic bit stuffing is used to be compatible to CAN FD. A transmitting node inserts after 5 consecutive bits of the same value a dynamic stuff bit with inverse value.
- **Fixed bit stuffing** – In the data phase, a fixed bit stuffing rule is used. A transmitting node inserts a fixed stuff bit after each 10th bit, counted from and including the DL1 bit. Accordingly, the stuff rate is S=11.

In the data phase, CAN XL has less stuff bits compared to CAN FD. This was done intentionally to increase net throughput. Table 1 shows the overhead due bit stuffing for all CAN types. In CC and FD, the worst-case ration between stuff bits and user bits is achieved with the following bit sequence. Letters `i` and `o` denote dynamic stuff bits.

```
bin: 0000 0i111 1o000 0i111 1o000 0i111 …
hex: 0    7     8     7     8     7     …
```

	worst case ratio stuff bit:user bits	**overhead**
CAN CC	1:4	1/(4+1)=20%
CAN FD	1:4	1/(4+1)=20%
CAN XL	1:10	1/(10+1)=9%

Table 1: Bit stuffing overhead in data field

Bit rate switching in CAN XL was carefully specified due to the high bit rate ratios to be switched, e.g. from 500 kbit/s to 20 Mbit/s). There are two changes in CAN XL compared to CAN FD.

- CAN XL switches the bit rate at the bit boundaries. (CAN FD switches the bit rate at the sample point.)
- CAN XL performs a hard synchronization after bit rate switching. This eliminates any phase error.

Concluding, in CAN XL bit rate switching has no impact on the frequency tolerance *df.*

C. Phase Error due to Frequency Tolerance

A node clock is not perfect. Therefore, its actual frequency f_{osc} is within a relative tolerance range of df around its nominal frequency f_{nom}.

$$f_{nom} \cdot (1 - df) \leq f_{osc} \leq f_{nom} \cdot (1 + df)$$

A bit time is an integer multiple of the CAN clock period. The consequence for the CAN nodes is that their absolute length of a bit time may slightly differ. This means the CAN nodes operate at slightly different bit rates.

Figure 1 illustrates the introduced phase error. To simplify the drawing, this example assumes all bus and transceiver delays to be zero. Delays would not change the phase errors but would shift the view of the receiver in the figure. f_{TX} is the transmitter's node clock frequency and f_{RX} the receiver's. Both clock sources have the same f_{nom}. The figure shows the case where $f_{RX} > f_{TX}$. This means the bit time in the receiving node is shorter than the bit time in the transmitting node. This difference leads to a positive phase error from the receiving node's point of view.

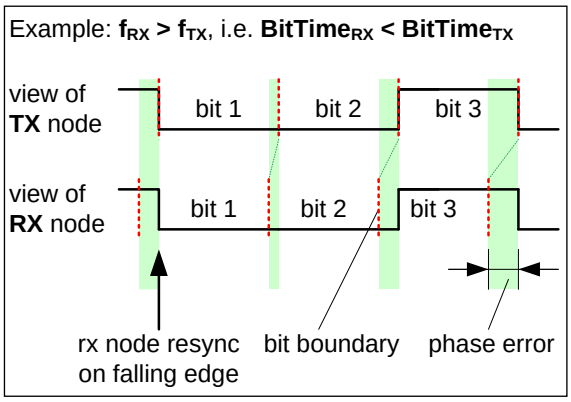

Figure 1: Phase error of receiving node due to frequency tolerance df

D. Mathematical background for derivation of df

Figure 2 shows the setup used to calculate the maximum clock frequency tolerance df.

We assume that the nominal node clock frequency f_{nom} is equal in both nodes, to simplify description. Since *df* is a relative tolerance, the results are also valid for the case, where the nominal node clock frequencies of the nodes differ.

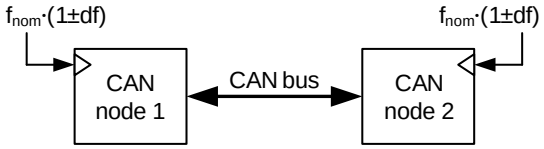

Figure 2: Setup for calculation of the accepted node clock frequency tolerance

The clock period *T* is the reciprocal of the frequency *f*.

$$f_{nom} = \frac{1}{T_{nom}}$$

When the frequency has a tolerance *df* the period changes in the following way:

$$f_{nom} \cdot (1 - df) = \frac{(1 - df)}{T_{nom}}$$

The worst-case scenario occurs when the receiving node uses the highest frequency $f_{RX} = f_{nom} \cdot (1 + df)$ and the transmitting node the lowest $f_{TX} = f_{nom} \cdot (1 - df)$. In this case the absolute duration of the synchronization jump width (SJW) in the receiving node $SJW_{nom} / (1 + df)$ is smaller than its nominal length.

We use the following simple inequality for the derivation of the *df* conditions.

$$x < \frac{\text{accepted phase error}}{\text{time to accumulate phase error}}$$

This inequality will lead to simple conditions, which are very good approximations of the exact df. In the following we proof that the approximated df is close enough to the exact df.

x is the relative difference of the clock periods of the two nodes: $T_{TX} = T_{RX} \cdot (1 + x)$. This is in line with the worst-case scenario mentioned, as it is $T_{RX} < T_{TX}$. Since we are interested in the frequency tolerance df and not in x, we have to calculate df from x. Figure 3 visualizes the relation between *df* and x.

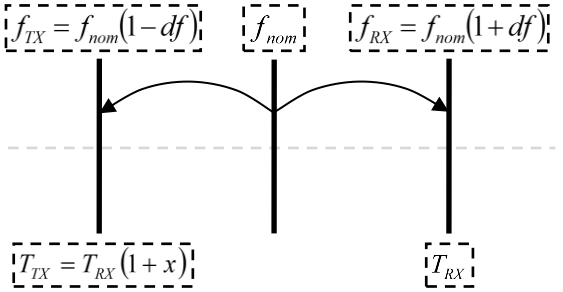

Figure 3: Relation between df and x

From Figure 3 we can derive *df* with the following steps.

$$T_{TX} = \frac{1}{f_{nom}(1 - df)}$$

$$T_{RX} = \frac{1}{f_{nom}(1 + df)}$$

$$T_{TX} = T_{RX}(1 + x) = \frac{(1 + x)}{f_{nom}(1 + df)}$$

$$T_{TX} = \frac{1}{f_{nom}(1 - df)} = \frac{(1 + x)}{f_{nom}(1 + df)}$$

From this follows that it is $df_{exact} = \frac{x}{2+x}$. Typically, x has a small value in the range of $0\% < x < 3\%$. This allows approximating *df* with high accuracy by $df_{approx} = \frac{x}{2}$. The absolute error introduced by the approximation is $df_{approx} - df_{exact} = \frac{x^2}{4+2x}$. This means for $x = 3$ % the absolute error is just 0.02 %. Concluding, we derive the

approximated conditions that are also listed in the ISO standard [12].

E. Definitions

This Subsection defines the variables used for the derivation of the conditions.

Subscripts
- A: variables of arb. phase
- D: variables of FD data phase
- X: variables of XL data phase

Elements of the bit timing configuration, without units.

BRP_D, BRP_A Bit Rate Prescaler

Elements of the bit timing configuration with seconds as unit:
- BT_A, BT_D, BT_X Bit Time
- TQ_A, TQ_D, TQ_X Time Quantum

Elements of the bit timing configuration with TQ as unit:
- sjw_A, sjw_D, sjw_X Sync. Jump Width
- $ps1_A$, $ps1_D$, $ps1_X$ Phase Segment 1
- $ps2_A$, $ps2_D$, $ps2_X$ Phase Segment 2
- bt_A, bt_D, bt_X Bit Time

F. Condition 1: Resynchronization (Arbitration Phase)

This condition ensures that a receiving node in the arbitration phase can eliminate the complete phase error by resynchronization. The arbitration phases in CAN CC, FD, and XL have the same properties. This condition is known since decades from ISO. Figure 4 shows the worst-case bit sequence. The worst-case distance between two falling edges is $10bt_A$.

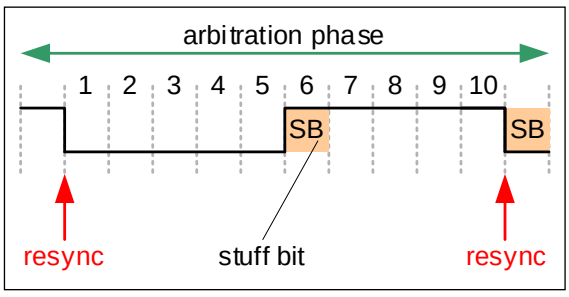

Figure 4: Worst-case bit sequence for resynchronization in the arbitration phase

A receiving node can reduce its phase error in the arbitration phase with each resynchronization by sjw_A. To be able to eliminate the complete phase error with each resynchronization, the following inequality must be met: $2\,df < \frac{sjw_A}{[10 \cdot bt_A]}$.

Condition 1 is:

$$df < \frac{sjw_A}{2 \cdot 10 \cdot bt_A}$$

G. Condition 2: Sampling Bit Succeeding own Error Flag

This condition ensures that a receiving node correctly samples the bit succeeding its own Error Flag. Thereby it can distinguish between local and global errors to correctly increment its receive error counter. This condition was derived decades ago for CAN CC. It also applies for CAN XL even if CAN XL always uses in the data phase a bit rate at least 2 times higher than in the arbitration phase.

Figure 5 shows the worst-case bit sequence for condition 2 in CAN CC.

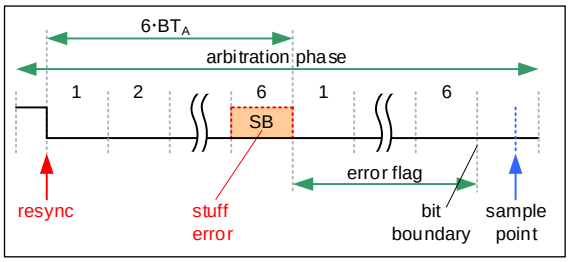

Figure 5: Worst case bit seq. for sampling bit succeeding own error flag in CAN CC

Figure 6 shows the worst-case bit sequence for condition 2 in CAN XL.

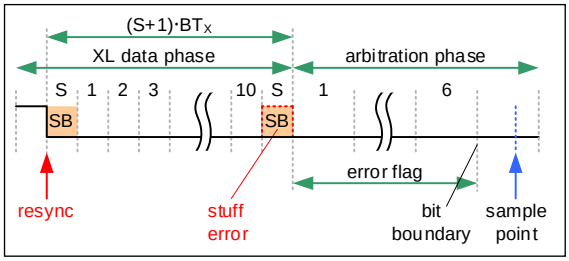

Figure 6: Worst-case bit seq. for sampling bit succeeding own error flag in CAN XL

When a CAN XL receiving node senses an error in the data phase, it switches back to the arbitration phase bit timing at the end of

the bit where it detects the protocol error. The node starts the transmission of its Error Flag in the subsequent bit. Since CAN XL receiving nodes do not use transceiver delay compensation (TDC), TDC does not need to be considered here.

The time from last resynchronization until the start of the error flag is:

- in CAN CC: 6*BT_A
- in CAN XL: (S+1)*BT_X=12*BT_X

In case of the minimal bit rate ratio of 2 (data/arbitration) in CAN XL, the time before the error flag is identical in CAN CC and XL. Concluding, condition 2 is a worst-case approximation for CAN XL nodes, since in case of bit rate ratios >2 a CAN XL node accepts a larger df than requested by condition 2.

During the time from last resynchronization until sampling the bit after its error flag, the phase error of the receiver has to be less than $min(ps1_A, ps2_A)$. This covers both cases: $f_{RX} < f_{TX}$ and $f_{RX} > f_{TX}$.

Condition 2 is:

$$df < \frac{\min(ps1_A, ps2_A)}{2 \cdot [13 \cdot bt_A - ps2_A]}$$

H. Condition 3: Resynchronization (XL Data Phase)

This condition ensures that a receiving node in the XL data phase can eliminate the complete phase error by resynchronization. Figure 7 shows a typical, but non-worst-case bit stream that can occur in the XL data phase of any CAN XL frame. Due to the stuff rate of S=11 a sequence of 11 bit with equal value is possible. Between two synchronizations here are 2*S=22 bit.

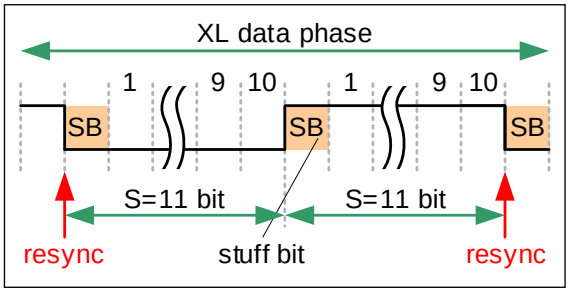

Figure 7: Non-worst-case bit sequence for resynchronization in the XL data phase

Figure 8 shows the worst-case bit sequence between two falling edges in the XL data phase. It can occur at the end of the XL data phase in frames with specific DLC values. Here the 10th bit in a row is missing, which would cause the insertion of a fixed stuff bit. The FCP field follows, where the fixed bit stuffing is not applied. The worst-case distance between two falling edges is: $(2S + 1) \cdot bt_X = 23 \cdot bt_X$
This bit sequence may only occur in XL frames with a data field size that is a multiple of 5 byte. As example it may occur in an XL frame with DLC=19 what means a data field size of 20 byte.

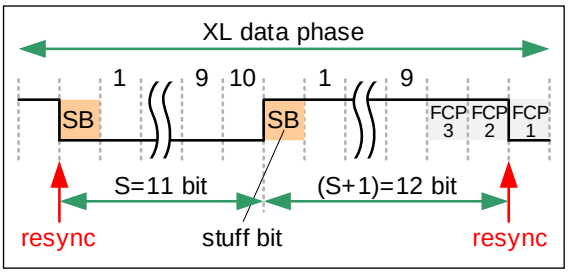

Figure 8: Worst-case bit sequence for resynchronization in the XL data phase

A CAN XL receiving node can reduce its phase error with each resynchronization by sjw_X. To be able to eliminate the complete phase error with each resynchronization, the following inequality has to be met: $2\,df < \frac{sjw_X}{[(2S+1) \cdot bt_X]}$.

Condition 3 is:

$$df < \frac{sjw_X}{2 \cdot (2S + 1) \cdot bt_X} = \frac{sjw_X}{46 \cdot bt_X}$$

3. Evaluation of Node Clock Frequency Tolerance (*df*)

A. Overview

This Section evaluates the node clock frequency tolerance *df* in CAN XL. The main target is to show how *df* depends on the bit rate and the type of transceiver used. Further we'll compare *df* in CAN FD and CAN XL.

B. Bit Timing Configurations for CAN XL

The conditions for *df* depend solely on the bit timing configuration. Therefore, we pick

several sets of bit timing configurations for our evaluation.

We limit the evaluation to the data phase and therefore use a single arbitration phase bit timing for all evaluations. We chose the arbitration phase bit rate to be 0.5 Mbit/s as this is a common bit rate. Table 2 shows the arbitration phase bit timing configuration.

Bit Rate [Mbit/s]	Sample Point	$BRP_A = BRP_X$	TQ_A / Bit	$prop\_seg_A$ [TQ_A]	$ps1_A$ [TQ_A]	$ps2_A$ [TQ_A]	sjw_A [TQ_A]
0.50	80%	1	320	191	64	64	64

*Table 2: **Arbitration phase** bit timing for **CAN XL** node, node clock = 160 MHz*

According [13], the optimal sample point (SP) position (part of bit timing) in the data phase depends on the physical layer (e.g., transceiver and network topology) [13]. The type of transceiver used has significant impact on the SP position. In general, we can say that the higher the symmetry of the transceiver, the closer is the optimal SP to the center of the bit. The transceiver types introduce different amounts (according to [13]) of bit asymmetry, listed here:

- HS CAN: large asymmetry
- FD 2 or 5: medium asymmetry
- SIC: small asymmetry
- SIC XL: very small asymmetry (when operated in FAST mode)

As exemplary XL data phase bit timing configurations we use:

- SIC XL transceiver optimized bit timings (values from CAN XL Plugfest in Baden Baden, Germany 2024), see Table 3;
- FD transceiver optimized bit timing, see Table 4 (means the SP positions are between 70 % and 80 %)

SIC transceivers have only a slightly worse asymmetry than SIC XL (operated in FAST mode) leading to only slightly higher SP values than with SIC XL transceivers. Therefore, a separate set of bit timings for SIC transceivers is not considered in this paper.

Bit Rate [Mbit/s]	Sample Point	$BRP_A = BRP_X$	TQ_X / Bit	$prop\_seg_X$ [TQ_X]	$ps1_X$ [TQ_X]	$ps2_X$ [TQ_X]	sjw_X [TQ_X]
2,0	51,3%	1	80	1	39	39	39
5,0	53,1%	1	32	1	15	15	15
8,0	55,0%	1	20	1	9	9	9
10,0	56,3%	1	16	1	7	7	7
12,3	53,8%	1	13	0	6	6	6
13,3	58,3%	1	12	1	5	5	5
14,5	54,5%	1	11	0	5	5	5
16,0	60,0%	1	10	1	4	4	4
17,7	55,6%	1	9	0	4	4	4
20,0	62,5%	1	8	1	3	3	3

*Table 3: **XL data phase** bit timing optimized for **SIC XL Transceivers (FAST mode)** for **CAN XL** node, node clock = 160 MHz*

Bit Rate [Mbit/s]	Sample Point	$BRP_A = BRP_X$	TQ_X / Bit	$prop\_seg_X$ [TQ_X]	$ps1_X$ [TQ_X]	$ps2_X$ [TQ_X]	sjw_X [TQ_X]
1,0	70,0%	1	160	63	48	48	48
2,0	75,0%	1	80	39	20	20	20
3,0	79,2%	1	53	30	11	11	11

*Table 4: **XL data phase** bit timing optimized for **FD Transceivers** for **CAN XL** node, node clock = 160 MHz*

Be aware that the given bit timing configurations are exemplary but considered to be realistic. Changing them will lead to other results.

C. Evaluation of df for CAN XL

Figure 9 shows the node clock frequency tolerance *df* for the bit timing configuration optimized for SIC XL transceivers (operated in FAST mode). It presents the result of each condition with a separate curve. The figure also shows the *df* range tolerated by a CAN XL node. The upper bound of *df* is determined by the lowest result from the 3 conditions.

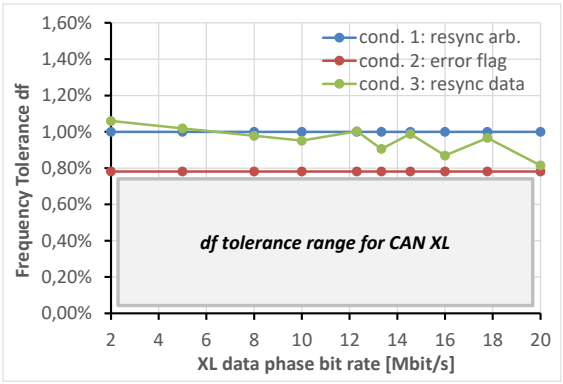

Figure 9: Frequency tolerance df in **CAN XL** based on bit timing optimized for **SIC XL Transceivers** (see Table 2 and Table 3)

Observations in Figure 9:

- All three conditions lead to roughly similar *df* values.
- df is independent from the bit rate.
- Condition 2 (error flag) from CAN CC limits the usable *df*. In other words, when using SIC XL transceivers in FAST mode the XL data phase is not limiting *df*. This is true even though the number of stuff bits was reduced by more than factor 2 (see Table 1). This did not happen by accident but was done intentionally during specification of CAN XL: We reduced the number of stuff bits to increase the net throughput, but we stopped at the point where it would have had a relevant impact on *df*.
- Condition 3 (resync. in data phase) leads to a df that is slightly decreasing towards higher bit rates. The reason for this is, that with increasing bit rate the number of time quanta per bit reduces. This forces us to configure the SP further away from the center of the bit.

Figure 10 shows the node clock frequency tolerance *df* for the bit timing optimized for FD transceivers. This is a valid use case where CAN XL is used at low speed of approx. 1 to 3 Mbit/s. The figure also shows the *df* range tolerated by a CAN XL node.

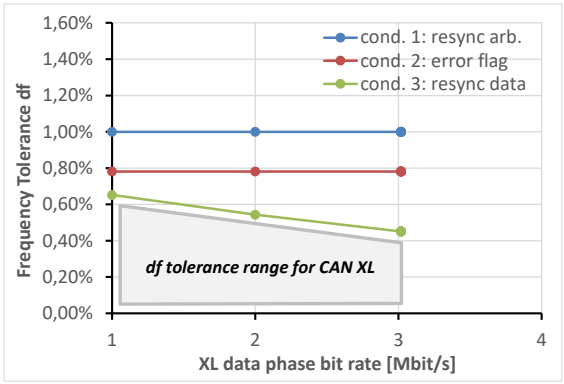

*Figure 10: Frequency tolerance df in **CAN XL** based on bit timing optimized for **FD Transceivers** (see Table 2 and Table 4)*

Observations in Figure 10:

- Condition 3 (resync. in data phase) is limiting *df*. Due to bit asymmetries resulting from the FD transceiver the SP has to be shifted towards the end of the bit. Since the asymmetries are absolute values, they impact high bit rates (short bit times) more than low ones.
- The absolute usable *df* of 0,4 % at 3 Mbit/s is still 2 to 3 times larger, than the typical *df* present in today's electronic control units (ECU).

D. Bit Timing Configurations for CAN FD

For the comparison of *df* in CAN XL to *df* in CAN FD we also define a set of bit timing configurations for legacy CAN FD nodes.

Bit Rate [Mbit/s]	**Sample Point**	**BRP_A**	**TQ_A / Bit**	**prop_seg$_A$ [TQ_A]**	**$ps1_A$ [TQ_A]**	**$ps2_A$ [TQ_A]**	**sjw_A TQ_A]**
0.50	80%	1	160	95	32	32	32

*Table 5: **Arbitration phase** bit timing for **CAN FD** node, node clock = 80 MHz*

Table 5 shows the arbitration phase bit timing. It has the same properties as the one for CAN XL, but it uses different parameters (necessary because FD nodes use lower frequencies and have a smaller configuration range for the time segments).

As exemplary FD data phase bit timing configurations we use a set with SPs optimized for SIC transceivers, see Table 6.

This means the SPs are close to the center of the bit, but not as close as it would be possible in CAN XL with SIC XL transceivers operated in FAST mode. Further, due to the limited configuration range for the time segments in CAN FD nodes we use for the low bit rates 1 and 2 Mbit/s a $BRP_D > 1$. This has no negative impact on *df*. (A negative impact would only occur if we would choose $BRP_D > BRP_A$.)

Bit Rate [Mbit/s]	Sample Point	BRP_D	TQ_D / Bit	$prop\_seg_D$ [TQ_D]	$ps1_D$ [TQ_D]	$ps2_D$ [TQ_D]	sjw_D TQ_D]
1,0	55,0%	4	20	1	9	9	9
2,0	55,0%	2	20	1	9	9	9
3,2	60,0%	1	25	4	10	10	10
5,0	56,3%	1	16	1	7	7	7
8,0	60,0%	1	10	1	4	4	4

*Table 6: **FD data phase** bit timing optimized for **SIC Transceivers** for **CAN FD** node, node clock = 80 MHz*

E. Comparison of df in CAN XL and CAN FD

Figure 11 shows the node clock frequency tolerance *df* of a CAN FD node with bit timing configuration optimized for SIC transceivers. The result of each condition is shown as a separate curve. The figure also shows the *df* range tolerated by a CAN FD node. The upper bound of *df* is determined by the lowest result from the 5 conditions of CAN FD.

Observations in Figure 11:

- Condition 3 (resync. in data phase) allows a twice as large *df* than all other conditions. This is due to the high number of stuff bits in CAN FD that lead to a falling edge latest after 10 bit. CAN XL uses in the data phase less then half the number of stuff bits.
- Condition 5 (bit rate switch) leads to a decreasing *df* towards higher bit rates. At 8 Mbit/s condition 5 is even the limiting one. (In use with FD transceivers, where the SP shifts towards the end of the bit, condition 5 would be at most bit rates the limiting one.)

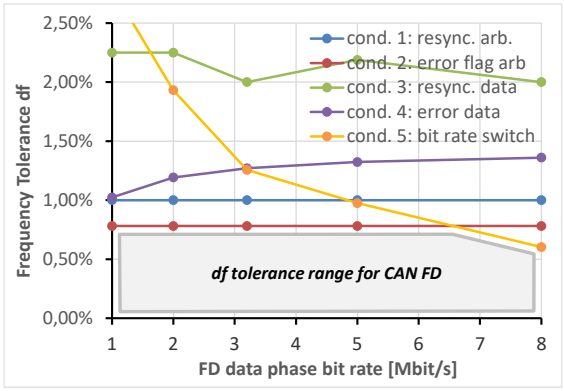

*Figure 11: Frequency tolerance df in **CAN FD** based on bit timing optimized for **SIC Transceivers** (see Table 5 and Table 6)*

Comparison of *df* in CAN FD (see Figure 11) and CAN XL (see Figure 9) for SIC and SIC XL Transceivers respectively, shows,

- that both, FD and XL, can use the same *df* range for the node clock despite the much higher bit rate in CAN XL.
- that the absolute max. value of *df* is 0,78%.
- that at high bit rates *df* of XL is even slightly larger than in FD.
- that in XL *df* does not depend on the bit rate ratio (data/arbitration), but in FD it does depend on it.

F. Summary of df Evaluation

The evaluation in this chapter has shown that the node clock frequency tolerance *df* is very similar for CAN FD or CAN XL nodes.

In general, we can say that using SIC or SIC XL transceivers (which cause only a low bit asymmetry) significantly improves *df*, by shifting the SP towards the center of the bit. In both, FD and XL, df is limited by condition 2 that comes from CC. Concluding, for the use with SIC and SIC XL transceivers, *df* in CC, FD and XL are the same. Consider that results might change when using other bit timing configurations.

4. Phase Margin Derivation

A. Overview

Non-accumulating errors are also present in a real system (e.g., bit symmetry error caused by the transceiver, cf. Section 1 C).

To be able to assess the robustness of a real CAN FD bus system, [5] and [6] introduce a metric called phase margin. In this Section we adapt this metric to the XL data phase.

B. Definition of Phase Margin

The phase margin (PM) is the allowed shift of a bit edge towards the SP of the bit, at a given precision of the node clock source frequency (df_{source}). In other words, this is the absolute edge shift caused by physical layer effects that is tolerated by a CAN node. Instead of a real edge shift, the RX signal can just be unreliable, e.g. due to ringing.

During frame transmission, the receiving nodes and transmitting nodes are sampling the bits from the CAN bus. The edge between two bits can be too early or too late. This leads to two PMs for the receiving node and two for the transmitting.

- PM1: Phase margin 1 of receiving node
- PM2: Phase margin 2 of receiving node
- PM1TX: Phase margin 1 of trans. node
- PM2TX: Phase margin 2 of trans. node

PM1TX and PM2TX are omitted in this paper because PM1 and PM2 are stricter when the Secondary Sample Point (SSP) in the transmitting node is chosen according to the following formula. [13] formally proofs this statement.

$$SSP = SP - \frac{1\ mTQ}{BRP \cdot TQ\ per\ bit\ time}$$

PM1 (before the SP) and PM2 (after the SP) help the system designer to evaluate whether the chosen data phase SP in the receiving nodes fits to the used physical layer. They are derived in the following.

C. Worst-Case Bit Sequence

For all four PMs, the worst-case bit sequence in XL data phase is when the transmitting node sends 11 bits with logical value ‘0’ followed by a fixed stuff bit with the logical value ‘1’. The bits after the fixed stuff bit don’t matter. The 11 bits with logical value ‘0’ occur when there is a fixed stuff bit with value ‘0’ followed by 10 regular bits with value ‘0’.

In case of High-Speed CAN, FD, and SIC transceivers, this is the longest possible sequence of dominant bits followed by a recessive bit inside a frame. Current transceiver designs cause the largest bit asymmetry at this bit sequence.

In case of SIC XL transceivers operated in FAST mode, this is the longest possible sequence of bits with the bit level ‘0’, followed by a bit with level ‘1’. Although a SIC XL transceiver behaves symmetric in the FAST mode, there is no alternative worst-case bit sequence with inverse polarity. This is since a CAN node synchronizes only on falling edges.

D. Phase Margin 1 of receiving node

Figure 12 shows the worst-case scenario for PM1.

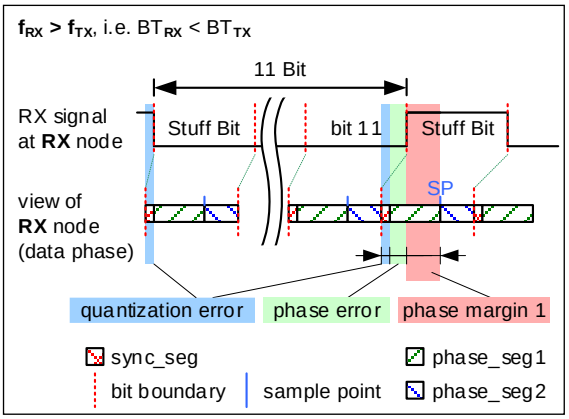

Figure 12: Phase Margin 1, worst-case bit sequence

The worst-case scenario for PM1 is when $f_{RX} > f_{TX}$ (due to *df*) and the quantization error is maximal. The higher CAN clock frequency of the receiving node leads to a phase error. This is a shift in the view of the receiving node, which means that the distance decreases between the SP of the stuff bit and the arriving edge at the beginning of this bit. The quantization error in the receiving node additionally shifts its view by one TQ towards the arriving edge.

To derive PM1 we need to calculate the difference between the following two times:

- time in receiving node from the incoming falling edge to the SP of the second stuff bit

- time of RX signal from falling to rising edge (11 bit)

The formula to calculate PM1 is:

$$PM1 = BRP_X \cdot T_{node_{clock}} \cdot \left(\frac{12 \cdot bt_X - ps2_X - 1}{(1 + df_{source})} - \frac{11 \cdot bt_X}{(1 - df_{source})}\right)$$

E. Phase Margin 2 of receiving node

Figure 13 shows the worst-case scenario for PM2.

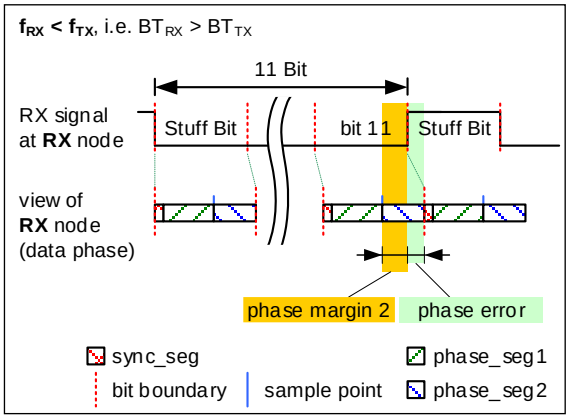

Figure 13: Phase Margin 2, worst-case bit sequence

The worst-case scenario for PM2 is when $f_{RX} < f_{TX}$ (due to df) and the quantization error is minimal. The lower CAN clock frequency of the receiving node leads to a phase error. This is a shift in the view of the receiving node, which means that the distance decreases between the SP of bit 11 and the arriving edge at the end of this bit.

To derive PM2 we need to calculate the difference between the following two times:

- time of RX signal from falling to rising edge (11 bit)
- time in receiving node from the incoming falling edge to the SP of bit 11

The formula to calculate PM2 is:

$$PM2 = BRP_X \cdot T_{node_{clock}} \cdot \left(\frac{11 \cdot bt_X}{(1 + df_{source})} - \frac{11 \cdot bt_X - ps2_X}{(1 - df_{source})}\right)$$

5. Evaluation of the Phase Margins

A. Overview

This Section evaluates the PM1 and PM2 derived in Section 4.

For the evaluation we used the set of XL data phase bit timings optimized for SIC XL transceivers (operated in FAST mode), see Table 3. This is the most interesting application as it supports the highest bit rates. The results for SIC transceivers will be similar since the bit timing configuration will be similar.

As we do not know the actual df_{source}, we calculate PM1 and PM2 for different values of df_{source} and draw a set of curves.

B. Phase Margin 1

Figure 14 shows PM1 for the mentioned exemplary XL data phase bit timings (see Section 5.A).

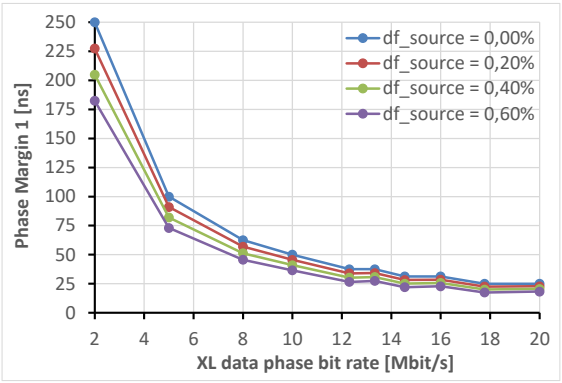

*Figure 14: **Phase margin 1** for **CAN XL** based on bit timing optimized for **SIC XL Transceivers** (see Table 2 and Table 3)*

Example how to interpret Figure 14 – Considering 2 Mbit/s and df_{source} = 0.2% a receiving CAN XL node can tolerate a 225 ns shift of the rising edge of the stuff bit towards the SP of the stuff bit. In other words, the RX signal can be unstable (e.g., due to ringing or asymmetries) during the first 225 ns of the stuff bit.

Observations

- PM1 decreases towards higher bit rates, proportionally to the bit time. E.g. PM1 = 225 ns at 2 Mbit/s and PM1 = 25 ns at 20 Mbit/s (approx. 10x less).

- PM1 highly depends on the SP position. A later SP increases PM1. In Figure 14 all SPs are roughly at 55 %, so this effect is not well visible.
- The impact of the frequency tolerance *df* on PM1 is small. This has two reasons. Firstly, the worst-case bit sequence for PM1 has just half of the length of the one in *df* condition 3. Secondly, the actual df_{source} in an implementation is typically much better than the maximally allowed *df*.

C. Phase Margin 2

Figure 15 shows PM2 for the mentioned exemplary XL data phase bit timings (see Section 5.A).

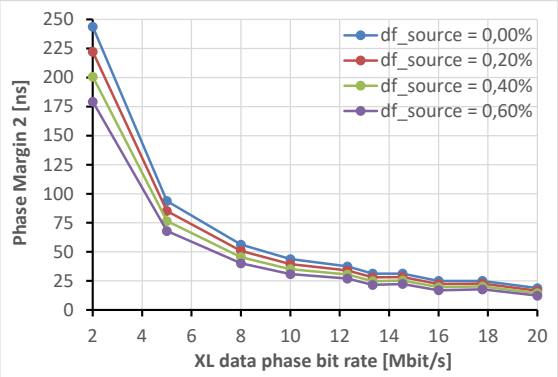

Figure 15: ***Phase margin 2*** *for* ***CAN XL*** *based on bit timing optimized for* ***SIC XL Transceivers*** *(see Table 2 and Table 3)*

The general observations are equal to the ones for PM1, except the one with the SP. A later sample point decreases PM2.

We also see in this example, that PM2 values are very similar to PM1. The reason for this is that the SP positions are close to the center of the bit. Since the SPs are around 55 % PM2 has slightly lower values than PM1.

As example, at 20 Mbit/s and df_{source} = 0,2 % PM2 is 17ns. This means the rising edge can occur 17 ns earlier than expected and the receiving node will still sample bit 11 correctly. These 17 ns are sufficient to cover the asymmetry introduced by the SIC XL transceiver of 10 ns and further 7 ns of asymmetry introduced by e.g. pins or clock jitter.

D. Summary

Both introduced phase margins extremely depend on the SP position. So, the SP positioning should be done by careful evaluation of the Physical Layer (Transceiver, network topology, etc.). We recommend a physical layer simulation and lab setup. The phase margin formulas are also provided in [13] along with an Excel sheet. [13] also helps the system designer to identify the asymmetries in his network.

6. Summary and Conclusion

Both, node clock frequency tolerance *df* and phase margin PM1/PM2 are two important metrics that must be used during CAN XL system design.

df and PM1/PM2 rely on different worst case bit sequences. The PM worst-case bit sequence depends on two different edges and targets asymmetries, while *df* uses the falling edge twice, so asymmetries are of no relevance.

The paper derived the 3 conditions to calculate the accepted *df* and evaluated *df* with several sets of bit timing configurations. Further it derived formulas for PM1 and PM2 and evaluated them with the same bit timings.

Both, PM and *df* depend on the SP position. This underlines the importance of the SP position for a robust CAN XL communication. The optimal SP position depends on the physical layer (e.g. transceiver, network topology). [13] provides help on this topic.

df, node clock frequency tolerance evaluation summary:

- CAN FD and CAN XL support the same *df* range when using SIC XL or SIC transceivers. CAN XL even supports a slightly larger *df* range. Results might change when using other bit timing configurations.
- SIC XL and SIC transceivers are beneficial for *df*, as they shift the SP towards the middle of the bit.

PM, phase margin evaluation summary:

- results mostly depend on SP position
- df has a minor impact on PM value

7. Acknowledgments

I want to thank Florian Hartwich for the valuable discussions around the topic of this paper.

Author	Dr. Arthur Mutter
Company	Robert Bosch GmbH
Address	Postfach 13 42, 72703 Reutlingen, Germany
Phone:	+49 7121 35-30684
E-mail:	arthur.mutter@de.bosch.com
Website:	http://www.can.bosch.com

References

[1] Uwe Kiencke, Siegfried Dais, Martin Litchel, “Automotive Serial Controller Area Network”, SAE Congress, Detroit, Michigan, USA, 1986

[2] Florian Hartwich, Armin Bassemir, “The Configuration of the CAN Bit Timing”, in Proceedings of the 6th international CAN Conference, Turin, Italy, 1999

[3] Florian Hartwich; “CAN with Flexible Data-Rate”, in Proceedings of the 13th international CAN Conference, Hambach Castle, Germany, 2012

[4] Florian Hartwich, “Bit Time Requirements for CAN FD”, in Proceedings of the 14th international CAN Conference, Paris, France, 2013

[5] Arthur Mutter; “Robustness of a CAN FD Bus System – About Oscillator Tolerance and Edge Deviations”, in Proceedings of the 14th international CAN Conference, Paris, France, 2013

[6] CiA601-3, “CAN FD node and system design, Part 3: System design recommendation”, Version 1.0.0, 2018

[7] Arthur Mutter and Florian Hartwich, “Advantages of CAN FD error detection mechanisms compared to classical CAN”, in Proceedings of the international CAN Conference, Vienna, Austria, 2015.

[8] Florian Hartwich, “Introducing CAN XL into CAN Networks”, in Proceedings of the 17th international CAN Conference, Baden Baden, Germany, 2020.

[9] Arthur Mutter, "CAN XL error detection capabilities", in Proceedings of the 17th international CAN Conference, Baden-Baden, Germany, 2020

[10] Christian Senger, “CRC Error Detection for CAN XL” in Proceedings of the 17th international CAN Conference, Baden Baden, Germany, 2020.

[11] Magnus-Maria. Hell, "The physical layer in the CAN XL world", in Proceedings of the 17th international CAN Conference, Baden-Baden, Germany, 2020.

[12] ISO 11898-1:2024, Road vehicles — Controller area network (CAN) — Part 1: Data link layer and physical coding sublayer

[13] CiA612-1, “CAN XL guidelines and application notes, Part 1: System design recommendations”, Version 0.0.9, 2023