# Robustness of a CAN FD Bus System – About Oscillator Tolerance and Edge Deviations

Dr. Arthur Mutter, Robert Bosch GmbH

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

**This paper derives a set of formulas to calculate the maximal accepted oscillator tolerance in CAN FD. Then it compares the oscillator tolerance of classical CAN and CAN FD. It shows that for realistic and non-extreme bit timing configurations CAN FD and classical CAN accept the same oscillator tolerance.**

**Furthermore, this paper introduces a metric called "phase margin" that allows to assess the robustness of a CAN FD bus system, i.e. up to which extent of physical layer effects the communication is reliable. Exemplary results show how this margin changes with the data phase bit rate.**

## 1. Introduction

One main target of a CAN bus designer is to achieve a reliable communication under all operating conditions on the bus. Therefore the designer has to (i) choose the suitable bit timing configuration, (ii) be aware of the accepted oscillator tolerance and choose the right oscillators and (iii) be aware of the effects on the bus (e.g. bit asymmetry, ringing).

CAN FD [2] [3] allows to use much shorter bit times than with classical CAN [1]. The shorter bit times of CAN FD require the designer to be more careful during bus design compared to classical CAN. For a robust bus communication, he has to choose the proper bit timing configuration and consider more analogue effects on the bus. The configuration of the CAN FD bit timing is explained in [5].

Classical CAN nodes according to ISO 11898-1 (2003) as well as CAN FD nodes synchronize on an incoming bit stream to enable the usage of low cost oscillators with less precision. The oscillator tolerance accepted by a CAN bus depends on the used bit timing configuration. CAN FD allows to send a portion of the CAN FD frame with a higher bit rate than the bit rate used for arbitration. The use of the higher bit rate requires switching the bit rate twice within a CAN FD frame. Due to this bit rate switching the formulas to calculate the accepted oscillator tolerance for classical CAN are not sufficient for CAN FD. In this paper, we derive the formulas to calculate the oscillator tolerance accepted by CAN FD. Further, we evaluate how the higher bit rate affects the oscillator tolerance.

The analogue effects on the bus cause absolute shifts of bit edges. Towards shorter bit times these effects have a larger impact on the robustness of the communication. The designer needs a metric to evaluate the robustness of a CAN FD bus. Therefore, this paper introduces a metric called phase margin and shows exemplary results.

### *A. Phase Error*

For each node, the distance (time) between the detected position and the expected position of an edge is called the phase error of that edge [4].

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

The error sources can be classified into two classes, which are independent of

each other: accumulating and non-accumulating errors.

**Accumulating Errors –** Different actual bit rates in sender and receiver lead to a phase error that adds up over time. Two errors lead to different bit rates:

- Frequency Error – This is caused by the frequency tolerance of the oscillator which feeds the CAN clock.
- Bit Rate Configuration Error – This is caused by a non-exact configured nominal bit rate in sender or receiver. E.g. a nominal bit rate of 1 Mbit/s is not configurable with a CAN clock frequency of 133.3 MHz. The non-exact nominal bit rate can be converted into a frequency error.

A CAN node (re)synchronizes repeatedly 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 errors 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 Rate Switch Error – In CAN FD bit rate switching takes place at the sample point. When sample point positions are different in the CAN nodes, the nodes switch the bit rate at different points during BRS and CRC DLM bit.
- Bit Symmetry Error – A bit is lengthened or shortened compared to how it was transmitted, i.e. the received bits are asymmetric. This is caused for example by
  - CAN transceiver
  - bus topology
  - electromagnetic disturbances
  - asymmetric rise and fall times of CAN RX and TX signals
  - jitter of oscillator frequency

The phase error introduced by non-accumulating errors doesn't need to be corrected via resynchronization.

The bus designer can completely eliminate two errors by using in all CAN FD nodes a CAN clock with the same frequency and the same bit timing configuration. These two errors are (i) bit rate configuration error and (ii) bit rate switch error. Therefore Bosch experts recommend to use just following CAN clock frequencies when implementing the protocol in hardware: 20, 40 or 80 MHz. Following this recommendation, the CAN FD node with the lowest CAN clock frequency limits the bit rate and bit timing settings, but all other CAN FD nodes can use identical settings.

### *B. Target of this Paper*

The target of this paper is to show how to deal with the two existing classes of errors. To deal with accumulating errors the paper derives the formulas to calculate the oscillator tolerance accepted in CAN FD. To deal with non-accumulating errors the paper derives a metric called phase margin. This is a measure for the total accepted edge shifts caused by non-accumulating errors.

## 2. Calculation of Oscillator Tolerance accepted in CAN FD

This Section derives a set of formulas to calculate the oscillator tolerance accepted by a CAN FD node. Each formula covers a worst case bit sequence. The Section recalls the formulas known from classical CAN and derives additional formulas to cover the bit rate switch in CAN FD.

These formulas allow calculating the theoretically accepted oscillator tolerance as a function of the CAN FD bit timing configuration. The formulas are inequalities and therefore conditions as well, i.e. the used oscillators' tolerance has to be smaller than the maximal result of the formula.

The formulas/conditions are derived based on worst case bit sequences which rarely happen. This means during normal operation (i.e. non-worst case) the system

typically can accept an oscillator tolerance that is higher than the calculated.

### A. CAN FD Properties

The most important properties of CAN FD, necessary to understand the derivation of the conditions 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. As example, if a receiving CAN node recognizes a recessive to dominant edge being earlier or later than expected, it corrects its internal view to be synchronous to the node transmitting the bits. Synchronization is performed only at recessive to dominant edges. [3] provides detailed description of the synchronization mechanisms in classic CAN. CAN FD uses identical mechanisms.

A CAN FD frame has two phases [2]

- 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 classical CAN.
- Data Phase – In the data phase only a single transmitter exists. In this phase the CAN FD protocol has no restrictions regarding bus length or maximum bit rate.

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 larger or equal to that in the arbitration phase.

The CAN FD bit stuffing rules are

- After 5 consecutive bits of the same value a stuff bit is inserted with inverse value.
- In the CRC field of the frame a fixed stuffing rule is used.
- Bit stuffing is independent of bit rate switching.

### B. Phase Error due to Tolerance of the Oscillator Frequency

An oscillator is not a perfect device. 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 the absolute length of a bit time may slightly differ from that of other nodes. This means the CAN nodes may 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 but shift the view of the receiver. $f_{TX}$ is the CAN clock frequency of the transmitting node and $f_{RX}$ is the one of the receiving node. Both nodes use 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 phase error from the receiving node's point of view.

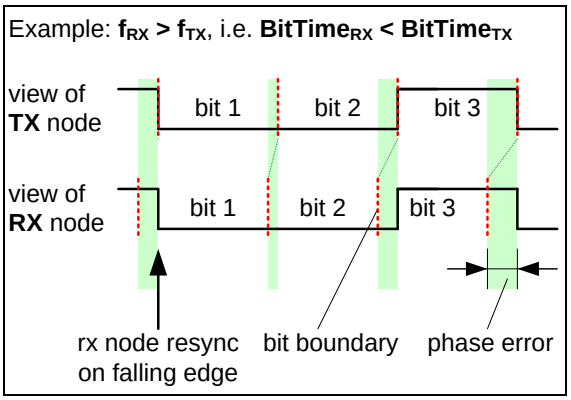


*Figure 1: Phase error of receiving node due to oscillator tolerance*

### C. Assumptions

Figure 2 shows the setup used to calculate the maximally accepted oscillator tolerance $df$ in CAN FD.

We assume that the nominal CAN 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 CAN clock frequencies of the nodes differ.

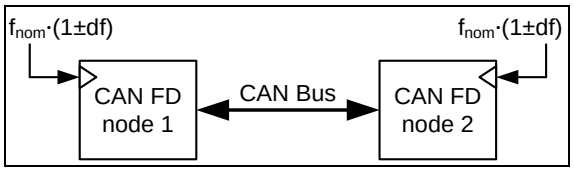


*Figure 2: Setup for calculation of the accepted oscillator tolerance in CAN FD*

For the calculations an ideal system is assumed, where the only source of error is the tolerance of the oscillator's frequency.

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 frequency $f_{TX} = f_{nom} \cdot (1-df)$. 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.

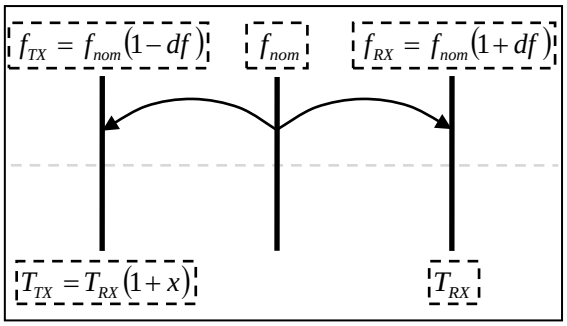


*Figure 3: Relation between $df$ and $x$*

To keep the conditions for the accepted oscillator tolerance simple we'll start their derivation with the following inequality

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

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

From Figure 3 follows: $df_{exact} = x/(2+x)$. Typically, $x$ is a small value in the range of $0\% < x < 3\%$. This allows approximating the accepted oscillator tolerance $df$ with high accuracy by $df_{approx} = x/2$. 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 0.022 %.

Since the absolute error introduced by approximation is negligible, we use this approximation to keep the conditions simple.

### *D. Definitions*

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

Subscripts

- A: denotes variables of arbitration phase, e.g. $TQ_A$
- D: denotes variables of Data Phase, e.g. $TQ_D$

Properties of the bit timing configuration with seconds as unit, nominal values

- $BRP_D$, $BRP_A$ Baud Rate Prescaler
- $SJW_D$, $SJW_A$ Synchr. Jump Width
- $PS1_D$, $PS1_A$ Phase Segment 1
- $PS2_D$, $PS2_A$ Phase Segment 2
- $BT_D$, $BT_A$ Bit Time duration
- $TQ_D$, $TQ_A$ Time Quantum

$$TQ_D = \frac{BRP_D}{f_{nom}}$$

$$TQ_A = \frac{BRP_A}{f_{nom}} = TQ_D \cdot \frac{BRP_A}{BRP_D}$$

Properties of the bit timing configuration with $TQ$ as unit:

- $sjw_D$, $sjw_A$ Synchr. Jump Width

- $ps1_D$, $ps1_A$ Phase Segment 1
- $ps2_D$, $ps2_A$ Phase Segment 2
- $bt_D$, $bt_A$ Bit Time duration
  e.g. $BT_D = bt_D \cdot TQ_D$

### E. Condition 1: Resynchronization (Arbitration Phase)

This condition ensures that a receiving node samples the bits in the arbitration phase correctly. The arbitration phase in CAN FD has the same properties like that of classical CAN. The corresponding condition is part of ISO 11898-1 (2003) [1] and is therefore not derived here:

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

### F. Condition 2: Sampling Bit Succeeding own Error Flag (Arbitration Phase)

This condition ensures that a receiving node correctly samples the bit succeeding its own Error Flag. Therewith it can distinguish between local and global errors to correctly increment its receive error counter. This condition is necessary during the arbitration phase, and the data phase when bit rate switching is not used. The corresponding condition is part of ISO 11898-1 (2003) [1] and is therefore not derived here:

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

### G. Condition 3: Resynchronization (Data Phase)

This condition ensures that a receiving node samples the bits in data phase correctly. A receiver resynchronizes on falling edges of the incoming bit stream. The worst case distance between two falling edges is $10 \cdot BT_D$. Figure 4 illustrates this worst case bit sequence.

A CAN FD receiving node can reduce its phase error with each resynchronization by $SJW_D$. To be able to eliminate the complete phase error with each resynchronization the following inequality has to be met: $2df < \frac{SJW_D}{10 \cdot BT_D}$.

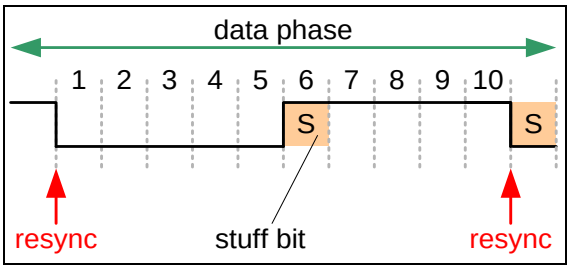


*Figure 4: Worst case bit sequence for resynchronization in the data phase*

From this follows the condition 3:

$$df < \frac{sjw_D}{2 \cdot 10 \cdot bt_D}$$

### H. Condition 4: Sampling Bit Succeeding own Error Flag (Data Phase)

This condition ensures that a receiving node correctly samples the bit succeeding its own Error Flag. Therewith it can distinguish between local and global errors to correctly increment its receive error counter.

When a CAN FD receiving node senses an error in the data phase, it switches back to the arbitration phase bit timing at the sample point where it detects the protocol error. The node starts the transmission of its Error Flag in the subsequent bit. Since CAN FD receiving nodes do not use transceiver delay compensation (TDC), TDC does not need to be considered here.

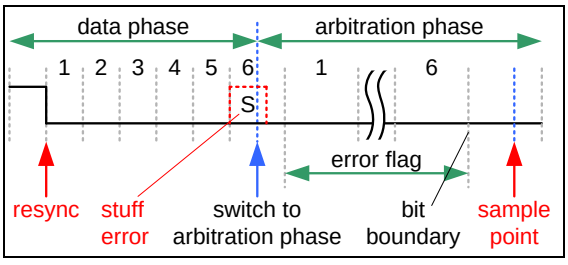


*Figure 5: Worst case bit sequence for sampling bit succeeding own Error Flag*

Figure 5 illustrates the worst case bit sequence. 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 $f_{RX} > f_{TX}$. Following inequality has to be met:

$$2df < \frac{\min(PS1_A, PS2_A)}{[(6 \cdot BT_D - PS2_D) + 7 \cdot BT_A]}.$$

From this follows condition 4:

$$df < \frac{\min(ps1_A, ps2_A)}{2 \cdot \left[ (6 \cdot bt_D - ps2_D) \cdot \frac{BRP_D}{BRP_A} + 7 \cdot bt_A \right]}$$

### *I. Condition 5: Switching from Arbitration Phase to Data Phase*

This condition ensures that a receiving node samples the first bits in data phase correctly. It is assumed that the bit rate is switched at the BRS bit. The worst case is the longest possible bit sequence without synchronization containing a bit rate switch from arbitration phase to data phase bit timing. Figure 6 illustrates the worst case bit sequence.

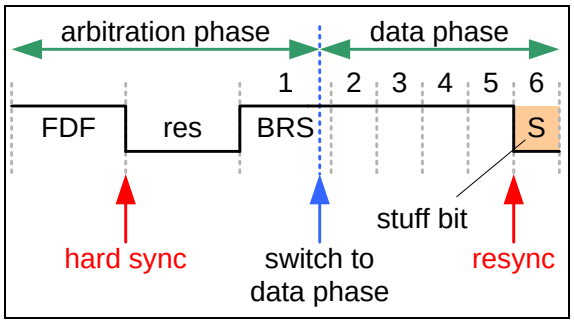


*Figure 6: Worst case bit sequence when switching from arbitration to data phase*

During switch over from arbitration phase to the data phase a part of the quantization error from the arbitration phase may be transformed into a phase error in the data phase. This transformation only occurs if a time quantum in arbitration phase is longer than a time quantum in the data phase: $TQ_A > TQ_D$.

Figure 7 shows an example where $TQ_A > TQ_D$. The view of the receiving node is not perfectly synchronous to the incoming RX signal. Its quantization error is $TQ_A$ and therewith maximal. After the bit rate switch the quantization error from arbitration phase is still present. The maximal possible quantization error in the data phase is $TQ_D$. Consequently, the remaining difference of quantization errors $TQ_A - TQ_D$ has to be considered in the data phase as a phase error.

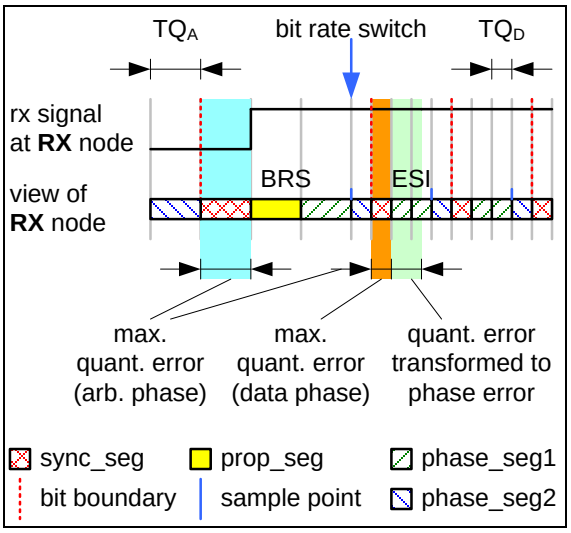


*Figure 7: Transformation of quantization error to phase error during bit rate switch*

If the user configures $TQ_A < TQ_D$, this difference gets negative. This means, that the receiving node has after bit rate switching, a lower quantization error in the data phase, than maximally allowed. For the case $f_{RX} < f_{TX}$ (due to oscillator tolerance) the worst case is when the quantization error is zero (minimal). Consequently, the difference has to be bounded to $\geq 0$.

From this follows, that the quantization error transformed with the bit rate switch into a phase error in the data phase in worst case is at most $\max(0; TQ_A - TQ_D)$.

According to the worst case bit sequence from Figure 6 the following inequality has to be met

$$2df < \frac{SJW_D - \max(0; TQ_A - TQ_D)}{[2 \cdot BT_A - PS2_A + PS2_D + 4 \cdot BT_D]}$$

From this follows condition 5:

$$df < \frac{sjw_D - \max\left(0;\ \frac{BRP_A}{BRP_D} - 1\right)}{2 \cdot \left[ (2 \cdot bt_A - ps2_A)\frac{BRP_A}{BRP_D} + ps2_D + 4 \cdot bt_D \right]}$$

### *J. Switching from Data Phase to Arbitration Phase*

After switching from data to arbitration phase, we have to ensure that a transmitting node can correctly sample the acknowledgement sent by the receiving nodes.

Derivation of the corresponding condition is lengthy and complicated, since receiving nodes drive the bus during the ACK bit. The outcome is, that switching from data to arbitration phase is much less critical than in previous cases in Section 2, i.e. the allowed tolerance of the oscillator frequency ( $df$ ) is much larger.

We argue verbally why this case is non-critical. Classical CAN already ensures a correct sampling of the acknowledgement (ACK bit). This means, that as long as bit rate switching does not introduce an additional phase error, CAN FD also ensures a correct sampling of the acknowledgement.

Bit rate switching can here only introduce an additional phase error, when $TQ_A < TQ_D$. This is an atypical case. The maximally introduced phase error is $TQ_D - TQ_A$ (cf. Section 2.I for a similar case). Two facts typically eliminate this additional phase error:

- CAN FD nodes receiving a CAN FD frame are synchronized more accurately just before the CRC DLM bit compared to classical CAN nodes receiving a classical CAN frame. This is due to the shorter bit times in the data phase and the fixed stuffing rule in the CRC field of the CAN FD frame.
- The transmitting node synchronizes on the recessive to dominant edge at the beginning of the ACK bit.

However, for some pathological bit timing configurations with $TQ_A << TQ_D$ switching from data to arbitration phase may get critical.

Remark: To increase robustness for the sampling of the ACK bit in CAN FD beyond classical CAN, CAN FD nodes tolerate two ACK bits or two CRC DLM bits. This improvement targets the case where non-accumulating errors lengthen the ACK bit so much, that it is sampled twice.

## 3. Evaluation of Oscillator Tolerance accepted in CAN FD

This Section evaluates the accepted oscillator tolerance in CAN FD. The main target is to show how the accepted oscillator tolerance of a CAN FD node changes towards higher data phase bit rates.

### *A. Assumptions*

As for the derivation of the conditions in Section 2 also for the evaluation an ideal system is assumed, where the only source of error is the tolerance of the oscillator frequency.

To calculate the accepted oscillator tolerance we need a set of CAN FD bit timing configurations. As we are interested in the effect of the data phase bit rate on the oscillator tolerance, we use the same arbitration phase bit timing configuration for all CAN FD bit timing configurations. Table 1 shows the arbitration phase bit timing configuration. We chose the arbitration phase bit rate to be 0.5 Mbit/s as this is today a common bit rate for classical CAN. Table 2 lists the data phase bit timing configurations. Parameters in both tables meet the value ranges allowed in the Bosch M_CAN IP module version 3.0.1.

We derived these bit timing configurations having two targets in mind: (i) it has to be a practical and realistic bit timing configuration and (ii) it should lead to a high value for the accepted oscillator tolerance. Ideally one would choose $BRP_D = BRP_A$. However, since we use just a single arbitration phase bit timing, this is not

possible. Consider that the given bit timing configurations are exemplary. Changing them will lead to other results.

| 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 | 80 | 47 | 16 | 16 | 16 |

*Table 1: Exemplary arbitration phase bit timing configuration, CAN clock = 40 MHz*

| Bit Rate [Mbit/s] | Sample Point | $BRP_D$ | $TQ_D$ / Bit | $Prop\_Seg_D$ [$TQ_D$] | $PS1_D$ [$TQ_D$] | $PS2_D$ [$TQ_D$] | $SJW_D$ $TQ_D$] |
|---|---|---|---|---|---|---|---|
| 1.00 | 60% | 4 | 10 | 0 | 5 | 4 | 4 |
| 1.25 | 63% | 4 | 8 | 0 | 4 | 3 | 3 |
| 1.67 | 67% | 2 | 12 | 0 | 7 | 4 | 4 |
| 1.82 | 64% | 2 | 11 | 0 | 6 | 4 | 4 |
| 2.00 | 60% | 2 | 10 | 0 | 5 | 4 | 4 |
| 2.22 | 67% | 2 | 9 | 0 | 5 | 3 | 3 |
| 2.50 | 63% | 2 | 8 | 0 | 4 | 3 | 3 |
| 2.86 | 64% | 1 | 14 | 0 | 8 | 5 | 4 |
| 3.33 | 67% | 1 | 12 | 0 | 7 | 4 | 4 |
| 3.64 | 64% | 1 | 11 | 0 | 6 | 4 | 4 |
| 4.00 | 60% | 1 | 10 | 0 | 5 | 4 | 4 |
| 4.44 | 67% | 1 | 9 | 0 | 5 | 3 | 3 |
| 5.00 | 75% | 1 | 8 | 0 | 5 | 2 | 2 |
| 6.67 | 67% | 1 | 6 | 0 | 3 | 2 | 2 |
| 8.00 | 80% | 1 | 5 | 0 | 3 | 1 | 1 |
| 10.00 | 75% | 1 | 4 | 0 | 2 | 1 | 1 |

*Table 2: Exemplary data phase bit timing configurations, CAN clock = 40 MHz*

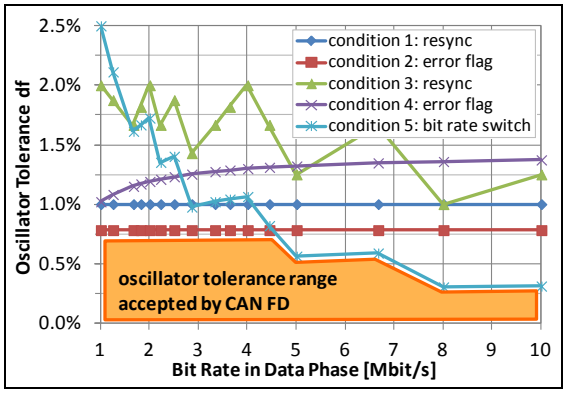


*Figure 8: Oscillator Tolerance accepted in CAN FD resulting from bit timing configurations in Table 1 and Table 2*

### *B. Evaluation*

Figure 8 shows the accepted oscillator tolerance resulting from the individual conditions in Section 2 for data phase bit rates of 1 to 10 Mbit/s. The Figure also shows the oscillator tolerance range accepted by a CAN FD node. This satisfies all 5 conditions.

Observations

- The classical CAN conditions (1 and 2) limit the accepted oscillator tolerance up to a ratio of data phase to arbitration phase bit rate of $4.44/0.5 \approx 9$.
- Condition 5 (bit rate switch) is the most critical condition at high data phase bit rates. Beyond a ratio of 9 of data phase to arbitration phase bit rate, condition 5 limits the accepted oscillator tolerance.

For other arbitration phase bit rates we made similar observations: up to a ratio of 7 to 10 of data phase to arbitration phase bit rate the classical CAN conditions limit the accepted oscillator tolerance.

### *C. Conclusion*

When the ratio of data phase bit rate to arbitration phase bit rate is not too large (e.g. $\leq 9$ with the bit timing configurations used in this example), CAN FD accepts the same oscillator tolerance as classical CAN. This assumes that the classical CAN uses the arbitration phase bit timing configuration of CAN FD.

From today's point of view the targeted data phase bit rates are in the range of 1 to 5 Mbit/s. For these bit rates CAN FD accepts the same oscillator tolerance like classical CAN.

## 4. Phase Margin

In a real system also non-accumulating errors (e.g. symmetry error, cf. Section 1.A) are present. To be able to assess the robustness of a real CAN FD bus system we define a metric called phase margin.

The phase margin is applicable to arbitration phase and data phase.

### *D. Definition*

In general, the phase margin is the allowed shift of a bit edge towards the sample point of the bit, at a given tolerance of the oscillator frequency. The absolute largest edge shifts are due to asymmetric delays caused by transceivers and bus topology.

A CAN node can correctly sample the incoming bits if the phase margin is larger than the maximal edge shift caused by all non-accumulating errors together.

In other words, the phase margin is the distance (time) between the sample point of a bit and the received edge of the ideal bit. Here, an ideal bit means a received bit that is not affected by any non-accumulating error. Since a received bit may have an edge at its beginning and at its end we need two phase margin definitions

- **Phase margin 1** is the distance (time) between the sample point of a bit and the received edge at the beginning of the ideal bit, at a given oscillator tolerance. Figure 9 visualizes phase margin 1.
- **Phase margin 2** is the distance (time) between the sample point of a bit and the received edge at the end of the ideal bit, at a given oscillator tolerance. Figure 10 visualizes phase margin 2.

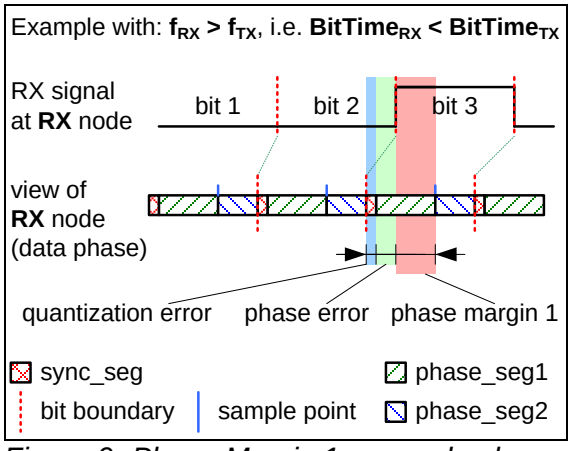


*Figure 9: Phase Margin 1, example shows data phase*

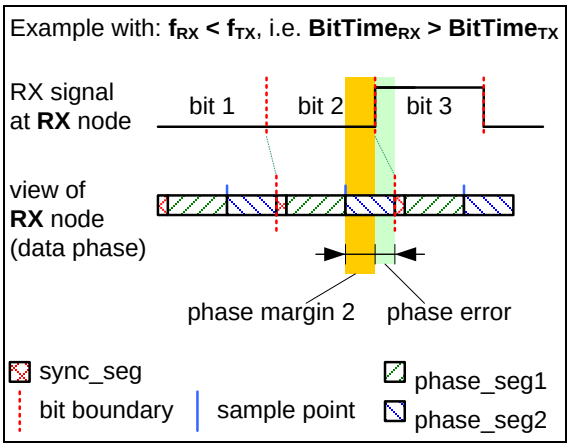


*Figure 10: Phase Margin 2, example shows data phase*

### *E. Phase Margin 1*

The worst case scenario for phase margin 1 is when $f_{RX} > f_{TX}$ (due to oscillator tolerance) and the quantization error is maximal. Figure 9 shows this case. 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 sample point of a 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.

Due to superposition of dominant bits during arbitration a dominant bit may get longer by at most the length of the propagation segment of the arbitration phase. This means, that the edge of a subsequent recessive bit may be shifted by the length of the propagation segment. Consequently, in arbitration phase the phase margin 1 has to additionally cover the propagation segment of the arbitration phase bit timing.

The sum of phase error and phase margin 1 is always

- in arbitration phase: $PS1_A + PropSeg_A$
- in data phase: $PS1_D$

Consequently, phase margin 1 is maximal when the oscillator tolerance is minimal ($df = 0\%$).

The phase error increases with the length of the considered bit sequence. Consequently, phase margin 1 decreases.

The formulas to calculate the phase margin 1 depend on worst case bit sequences. Since edge shifts are caused by physical layer effects, the worst bit sequences have to be found together with physical layer experts.

### *F. Phase Margin 2*

The worst case scenario for phase margin 2 is when $f_{RX} < f_{TX}$ (due to oscillator tolerance) and the quantization error is minimal. Figure 10 shows this case. 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 sample point of a bit and the arriving edge at the end of this bit.

The sum of phase error and phase margin 2 is always

- in arbitration phase: $PS2_A$
- in data phase: $PS2_D$

Similar to phase margin 1, phase margin 2 is maximal when the oscillator tolerance is minimal ($df = 0\%$).

The phase error increases with the length of the considered bit sequence. Consequently, phase margin 2 decreases.

Similar to phase margin 1, the worst bit sequences for phase margin 2 have to be found together with physical layer experts.

## 5. Evaluation of the Phase Margins

This Section evaluates the phase margins defined in Section 4. Evaluation is just exemplary as the worst case bit sequences have to be discussed with the physical layer experts. Further, we limit evaluation to the data phase, as this is more critical than the arbitration phase.

Exemplarily, we evaluate the case where the CAN transceiver causes the largest bit asymmetry. This case may turn out to be the worst or at least one of the worst. Therefore the transmitting node sends 5 dominant bits followed by one recessive stuff bit. Figure 11 shows the two phase margins to be evaluated in this case. The Figure assumes $df = 0\%$ to simplify the drawing.

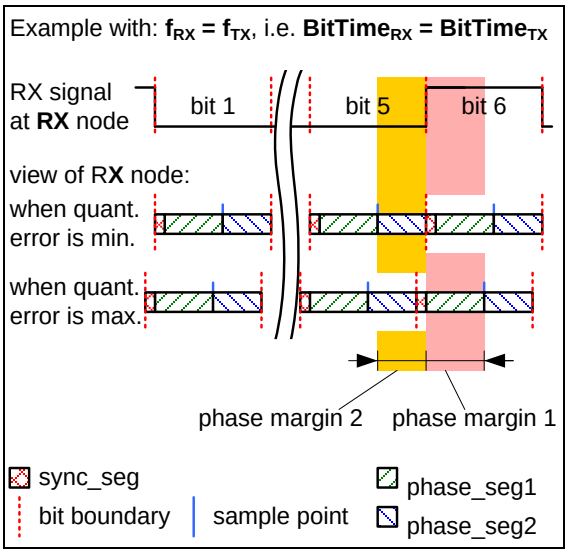


*Figure 11: Relevant phase margins for the case where a transmitter sends 5 dominant bits and one recessive stuff bit*

We used the bit timing configurations from Table 2 to calculate phase margin 1 and 2.

### *A. Phase Margin 1*

Figure 12 shows phase margin 1 (left y-axis) for data phase bit rates of 1 to 10 Mbit/s. Phase margin 1 is drawn for different oscillator tolerances leading to a set of curves. Additionally, the Figure shows the nominal sample point position (right y-axis).

General observations

- Phase margin 1 decreases towards higher bit rates, because the bit time decreases.
- The impact of the frequency tolerance of the used oscillator ($df_{used}$) is small, because in the considered case the bit sequence of roughly 5 data phase bits has an absolute short duration.
- A later sample point position increases phase margin 1. As example, the phase margins at 4 and at 5 Mbit/s are

nearly equal, because the sample point at 5 Mbit/s is later.

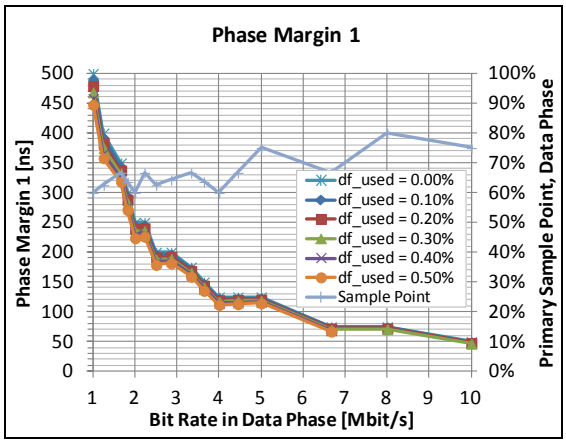


*Figure 12: Phase margin 1 for exemplary considered case*

The first generation of CAN FD transceivers will be qualified for 2 Mbit/s. These transceivers are planned to provide in the case considered in this example a recessive bit length of 400 ns to 550 ns at 2 Mbit/s (nominal bit time 500 ns) [6]. These values are only valid without a bus connected.

At 2 Mbit/s and e.g. $df_{used} = 0.5\%$ phase margin 1 is 224 ns. This means it is sufficient to detect a recessive bus level 224 ns after the start of the bit in the receiving node. This is enough to tolerate up to 100 ns delay introduced by the transceiver and concurrently further 124 ns introduced by other reasons, e.g. due to bus topology (high capacitive load on the bus) or jitter caused by RF fields.

### *B. Phase Margin 2*

Figure 13 shows phase margin 2 (left y-axis) for data phase bit rates of 1 to 10 Mbit/s. Phase margin 2 is drawn for different oscillator tolerances leading to a set of curves. Additionally, the Figure shows the sample point position (right y-axis).

The general observations are equal to the ones for phase margin 1, except the observation with the sample point. A later sample point decreases phase margin 2.

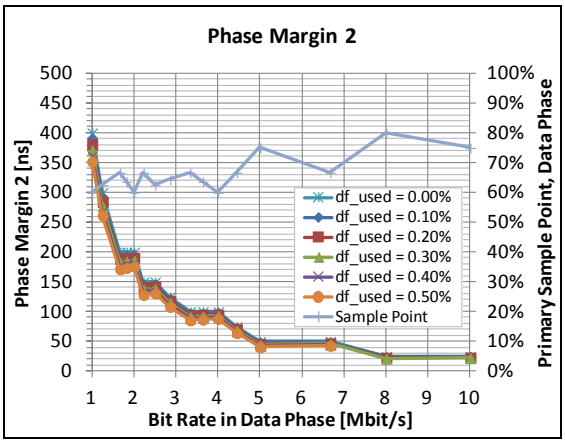


*Figure 13: Phase margin 2 for exemplary considered case*

At 2 Mbit/s and e.g. $df_{used} = 0.5\%$ phase margin 2 is 176 ns. This means, a receiving node samples bit 5 correctly even if it detects a recessive bus level already 176 ns before the end of bit 5. This is enough to tolerate up to 50 ns asymmetry introduced by the transceiver and concurrently further 126 ns introduced by other reasons.

### *C. Summary*

Both introduced phase margins depend on the bit timing configuration and the considered bit sequence. Here we showed an exemplary evaluation as the worst case bit sequences still have to be discussed with CAN physical layer experts.

A general observation is that both phase margins decrease quickly towards increasing bit rates, i.e. the phase margins are proportional to the bit time. Due to this, a bus designer has to carefully consider all errors in its bus system (cf. Section 1.A). We recommend a physical layer simulation of the bus system and a lab setup to quantify the physical layer effects.

## 6. Summary and Conclusion

As a first step error sources leading to a phase error in a CAN FD bus system were classified into two classes: accumulating errors and non-accumulating errors. To be able to quantify the robustness of a CAN FD bus system with respect to both

error classes, we provide one method for each class.

This paper derives 5 conditions that have to hold for the frequency tolerance (accumulating error) of the oscillator used in a CAN FD bus system. Evaluation shows that for non-extreme bit timing configurations – e.g. ratio of data to arbitration phase bit rate < 9 – CAN FD and classical CAN accept the same oscillator tolerance.

Non-accumulating errors (e.g. bit symmetry error due to physical layer effects) are either not or just slightly bit rate dependent. To assess the robustness of a CAN FD bus system with respect to non-accumulating errors, this paper defines a metric called phase margin. Exemplary evaluation shows that the phase margin decreases proportionally with decreasing bit times.

Both, accepted oscillator tolerance and phase margin, have to be considered during a CAN FD bus system design, as they rely on different worst case bit sequences. Further, as both depend on the bit timing configuration, a bus designer should optimize the bit timing configuration for the particular bus system.

For a successful CAN FD bus design we recommend – beside the theoretical evaluations shown in this paper – to additionally simulate and set up the bus topology in the lab. The analog simulation should contain models for the used transceivers types, common mode chokes, cabling and terminations. This will help quantifying the physical layer effects.

## 7. Acknowledgments

I want to thank the following people for the valuable discussions around the topic of this paper: Sascha Mäusling (Daimler), Dr. Marc Schreiner (Daimler) and Bernd Elend (NXP).

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

**References**

[1] ISO 11898-1 (2003); Road vehicles — Controller area network (CAN) — Part 1: Data link layer and physical signalling

[2] Robert Bosch GmbH; CAN with Flexible Data-Rate; Specification; Version 1.0; Date 17.04.2012, http://www.bosch-semiconductors.de/media/pdf_1/canliteratur/can_fd_spec.pdf

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

[4] Florian Hartwich; The Configuration of the CAN Bit Timing; Proceedings of the 6th international CAN Conference, Turin, Italy, 1999

[5] Florian Hartwich; The Configuration of the CAN FD Bit Timing; Proceedings of the 14th international CAN Conference, Paris, France, 2013

[6] CAN in Automation (CiA); (Work Draft) CAN FD – Part 1: Physical interface implementation, Version 0.0.2, 1st June 2013