Improved architecture for Profinet IRT devices

Christoph Felser, Max Felser
Bern University Applied Science
Jlcoweg 1
3400 Burgdorf - CH
christoph.felser@bfh.ch, max.felser@bfh.ch

Hassan Kaghazchi
University of Limerick
hassan.kaghazchi@ul.ie

Abstract

This paper deals with Real-Time applications based on Profinet IO Real-Time Class 3. The industrial automation in some cases requires the precision of IO data timing well below the delivered cycle time on the bus. When using the Isochronous Real-Time protocol of Profinet, one single predefined time within a cycle time can be chosen for the inputs- and outputs. The work presented here, shows how the architecture of a Profinet device can be improved. The user can define the time of an input or output within the given time slot with an accuracy of 65 nanoseconds. Furthermore the feasibility of the proposed design is proven with several measurements and simulations.

1. Introduction

Modern control systems use industrial communications networks for the distribution of control functionality over different physical controllers. If the control function has to meet the real-time constraints, the network has to ensure minimal quality of service in relation to the timing and synchronization of the transmitted data.

The International Electrotechnical Commission (IEC) defines a set of solutions for fieldbus and Real-Time Ethernet (RTE) based networks in IEC 61158 [1] and IEC 61784 [2], [3]. But not all of these solutions are able to cope with hard real-time requirements [4].

In this paper a possible set of Performance Indicators (PI) will be defined, and it will be shown using examples how these indicators are met by such networks.

In a second clause the structure of a Profinet IO system using the Isochronous Real-Time (IRT) protocol for the exchange of cyclic data is outlined and the results of some measurements taken from a practical setup are provided.

In the last clause an extension to the application interface of a Profinet IRT system is provided, and it will be shown how the requirement of the benchmark with this solution can be met.

2. Performance Indicators

A number of industrial networks are defined in IEC standards and are implemented by different manufacturers in their automation devices. The user needs a mechanism to compare the achievable performance of these different networks and match it with the requirements of the distributed control application.

2.1. Definitions

In [5] the author distinguishes the process data from configurations, parameters and programs. Here only the process data is considered. For this the absolute temporal consistency, the relative temporal consistency and the spatial consistency have to be ensured.

In [3] the following Performance Indications (PI) are defined: delivery time, number of end nodes, basic network topology, number of switches between end nodes, throughput of RTE, non-RTE bandwidth, time synchronization accuracy and redundancy recovery time. In this case only two of these process data timings are considered as relevant PIs; namely the delivery time and the time synchronization accuracy.

Delivery time is the time needed to convey a Service Data Unit (SDU, message payload) from one node (source) to another node (destination). The delivery time is measured at the Application Layer interface. The maximum delivery time shall be stated for the two cases of no transmission errors, and one lost frame with recovery.

The time synchronization accuracy shall indicate the maximum deviation between any two node clocks.

2.2. Published measurements

Different research groups have already published measurements of relative temporal consistency for different field buses or RTE solutions. Here some examples which are in direct competition to Profinet systems are used.

In [6] a square wave output signal with a 3 ms cycle time is produced using a Profibus DP-V2 isochronous system. The measurements show that with a synchronous application there is no visible jitter. However no indication of the actual value of the jitter is provided in this work.

In [7] & [8] a square wave signal with 1ms cycle is used as an input to an EtherCAT system and the temporal consistency of different inputs is measured with the help of the Distributed Clock (DC) system of EtherCAT bus. The measurements show, that the device can reach a deviation of less than 13ns with a standard deviation in the order of 4 ns.

For Profinet different Real Time Classes (RTC) can be differentiated. The Real-Time Class 1 (RTC1) is the best effort transmission; Real-Time class 3 is Isochronous Real-Time (IRT) communication with the possibility to synchronize also the application cycle [9].

The RTC1 was measured in [10] by generating a square wave output signal with a 16ms cycle. The resulting jitter in this experimental setup reached up to 5% of the cycle time.

For RTC3 frames square wave output signals with cycle times of 1, 2, 3, and 4 ms were generated [11]. The jitter was evaluated both for the frame itself and for the output signal using the ERTEC ASIC. The results show that the RTC3 frames have a reasonable jitter of less than 135 ns with a standard deviation in the order of 20 ns. But the application output jitter is in the order of 100 µs. This paper shows how the jitter may be reduced down to several nanoseconds.

2.3. Simple Benchmark

To get an idea of the usefulness of the different Performance Indicators, a simple benchmark system is defined. The proposed system must be simple enough to setup in a laboratory environment for testing, yet at the same time have timing requirements which are typical in Real-Time applications in practice.

For this the “turning arrow” system is taken. On a turning disc an arrow, is marked. A stroboscopic flash gives the impression to the human eye that the arrow is stationary or is moving with a lower speed than the speed of the disc (fig. 1).

As an interface to the distributed control system a binary sensor detects one turn of the disc and generates a signal at this event (t_{index}). In addition, a digital output has to activate the flash at the correct time (t_{strobe}).

The index is generated at the t_{index} and transmitted with the delay of $t_{input\text{-}delivery}$ to the controller. The controller measures the Δt_{index} and defines the exact position of t_{strobe}. This actuation is transmitted with the delay of $t_{output\text{-}delivery}$ to the second device which sets the corresponding output.

The value of Δt_{index} depends on the RPM of the disc and varies in the range of 1 to 12,000 RPM, hence it takes 138ns to turn the disc by 0.01 degree at the maximum speed. If a jitter of 0.01 degree is considered as acceptable, then the sum of all delays must be less than 138 ns.

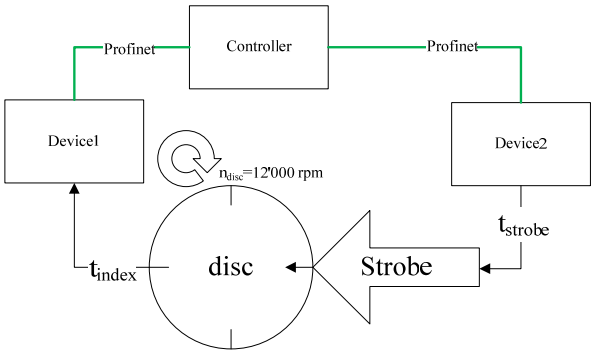

Figure 1. "turning arrow" system

With this benchmark setup the Performance Indicator “Delivery Time” is not important, as long as the delivery time is known or has always the same value. This means a well limited jitter which must be less than the 138 ns for this application.

3. Different architectures of Profinet devices

It is known that Profinet is very fast at the bus level [11], however it is pointless to have a very accurate bus, if it is not possible to transmit the data from the bus level to the application level with the same precision. Depending on the design or the architecture of a Profinet device, the required accuracy could be achieved. Therefore the most common designs are reviewed and their performance and accuracy is estimated. To increase the delivered precision, an improvement of the design is presented.

In most cases the Profinet communication stack works independent of the application. Often they communicate over shared memory with each other. This means that the application polls the data with a certain cycle time. Consequently the cycles of the application and of the communication stack work in an asynchronous manner. The maximum deviation or jitter equals to the polling cycle time. As a jitter below one microsecond is required, it should nearly be impossible to build a software based application, which meets this requirement. Hence this architecture is not suitable for IRT communication.

In some network controllers both the application and the communication stack operate simultaneously in the same controller. With the usage of a preemptive real-time operating system, it is possible to run different tasks concurrently. As the application and the Profinet stack operate in the same operating system as different tasks, the shared memory is no longer required and a more direct communication can take

place. Instead of polling the data, the application can register a callback function. The later is invoked by the Profinet stack if new cyclic data comes in. In this way the application and the stack operate in a synchronous manner. Even a preemptive real-time operating system in some cases needs an unpredictable amount of time to switch from one task to another or to service an interrupt routine. Of course this depends also on the hardware platform, the performance of the operating system and the priorities of the different tasks. But it is nearly impossible to find a reasonable OS that offers a smaller jitter than one microsecond.

The third possibility is that the stack and the application can work independently on two different network controllers. Instead of polling the data in the shared memory, the stack can procure a signal, which requests an interrupt routine in the application. In this routine the application reads the current data out of the shared memory. In this case the jitter depends on the accuracy of the procured signal of the stack and the interrupt latency in the application. A typical real-time kernel used for software based architecture, yields an interrupt-latency with a jitter of several microseconds and hence doesn't meet the required accuracy for an isochronous Profinet application.

The last option for a Profinet device is very similar to the third one. The software is replaced with a hardware-based application. In such a scenario the jitter is reduced to the clock cycle in the hardware and the procured signal from the communication stack. When a clock above one megahertz is used, the application itself delivers the required precision for IRT communication with Profinet (see clause 2.3). As the hardware-based application is much less flexible, the complex tasks maybe difficult to realize. Therefore, sometimes it is unavoidable to use a software based application.

The developers of network controllers know the issue of software based solutions and often procure a hardware based synchronization output signal. The signal marks the time when the received cyclic data are ready to be used, but doesn't represent the time when the packet actually arrives on the Ethernet [12][13]. Otherwise it would be impossible to synchronize the different devices in the network. Even if there is a very accurate cycle time on the bus, the cyclic data never arrives exactly at the same time in each device. That's why the cycle time in the stack of the devices is synchronized over the network. The synchronization mechanism of Profinet is well explained in [14].

As the second and third architectures described above are both work in a similar fashion, this paper focuses only on the second architecture. The following proposition to improve the precision works with both architectures and uses the hardware based synchronization output signal. The last option described doesn't need to be improved, because it already meets all the requirements.

3.1. Improving the accuracy

To decrease the jitter between the application and the stack, a hardware based latch is added to the set-up (fig. 2).

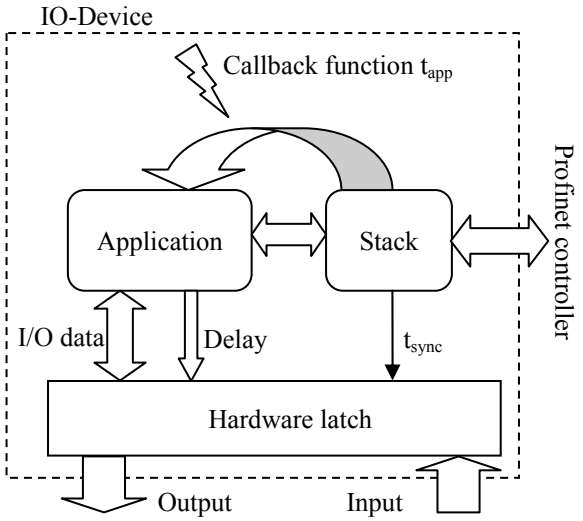

Figure 2. Improved Architecture

The synchronization output signal (t_{sync}) of the network controller triggers the start of the preset delay (t_{delay}) in this latch. Within the set delay time of the hardware latch, the stack invokes the callback function, in which the cyclic data are processed and passed to the added hardware. As soon as t_{delay} expires, the hardware latch transfers the data to the final output and reads the input at the same time (t_{io}). It is also possible that the delay time is transmitted over Profinet and set by the application. Depending on the resolution of the hardware timer in the latch (t_{clock}), the I/O data can be read or written in a very accurate way. Since the primary goal of the delay is to eliminate the inconsistent delay of the callback function, the hardware produced delay can never be smaller than the maximum delay time between the synchronization signal and the output of the application (Δt_{app}). Nevertheless, if the outputs are not written in each available cycle, the whole timeslot can be used in the given resolution. When the I/O is read or written with a delay smaller than the maximum delay between the synchronization and the application output, the timer should start one cycle before and with a delay of the cycle time plus the demanded delay (fig. 3). This way the exact time of the output or input can be chosen. In contrast to the solution presented in the IEC 61158-5-10 clause 8.3.7 [1], with this solution the whole given timeslot can be used. According to the standard the delay has to be constant, hence all the devices write or read the outputs and the inputs at the same time. With

the proposed architecture the devices are still synchronized, but the user can choose the exact time for the outputs and the inputs in each device, by transmitting the appropriate delay over Profinet to the different devices. Hence the hardware latch delay can be changed in each cycle. The resultant jitter of the IO equals to the sum of the jitters on the synchronized output signal and the jitter in the hardware latch.

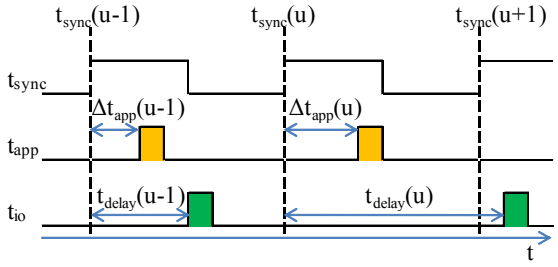

Figure 3. Timing Characteristic

Pretty much the same can be done to realize a very accurate synchronized input impulse for Profinet devices with the callback function as an application trigger (fig. 4). The synchronization signal from the network controller is used to synchronize the timer in the added hardware latch (t_{timer}). The timer in the latch counts from zero to the set cycle time of the bus i.e. one millisecond. To detect the instant of time of an impulse at the input (t_{imp}), the hardware latch timestamps the start of the impulse. In every cycle the application reads the results from the last cycle out of the latch. As a result the input data and the corresponding timestamp will be transferred from the latch to the application. Depending on the resolution of the timer in the latch, a very high accuracy can be realized.

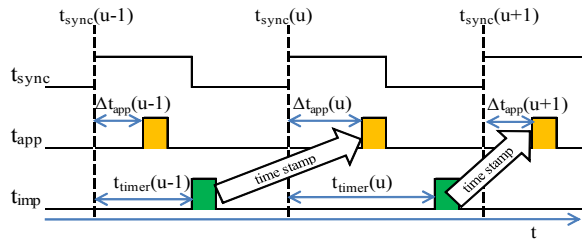

Figure 4. Timing of an impulse

3.2. Measured performance of different architectures

A very simple Profinet network is built, in which the device exchanges cyclic data every millisecond. As a Profinet IO controller CPU315-F2 PN/DP from Siemens was used and as the IO device the development board for the network controller from Hilscher (netX500). To run the application and the Profinet device stack on the netX in different tasks simultaneously, the preemptive real-time operation system from Hilscher (rcX) is engaged. As soon as new cyclic data arrives, a callback function is invoked, which processes them. This architecture equals to the second option presented in the previous section of this paper. The configuration part the callback function represents the application itself. In every cycle the application simply generates a falling edge at the general purpose output of the development board, by setting the output first to zero and then back to one. In this way the exact time for the callback function and its duration is determined.

First the cycle times on the bus were measured with an Ethernet-Tap from Hilscher [15], which has a resolution of ten nanoseconds. A user-written program analyzed the delivered pcap files and returned the cycle times of the different participants on the bus.

The resultant distribution of cycle times is in line with the similar measurements carried out by other reported work and establishes the accuracy of the Profinet packets on the bus. In 10,000 measured cycles the maximum deviation is within 80 nanoseconds. But as mentioned above, these values don't necessarily represent the precision of the final output of the system. The accuracy of the cyclic transferred data only influences the delay between the times when the data are to be used and when they are ready to be read. Therefore in a system with software based applications, the delay caused by the bus is much smaller than the delay in the application itself and can be omitted.

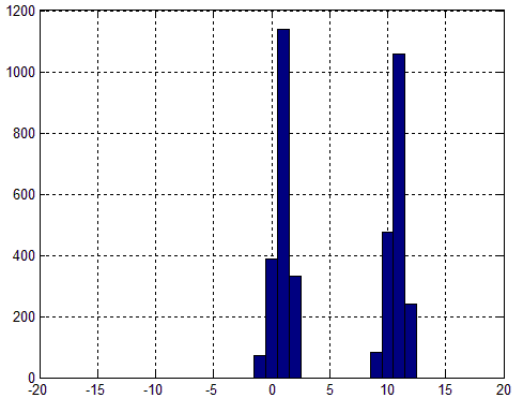

Figure 5. Distribution of δt_{sync} (temporal) in ns

The netX also enables the synchronization of the output signal when using the RTC3 transmission. The accuracy of the proposed architecture is dependent on the precision of this synchronization output signal and the precision of the added hardware latch. To measure the performance of the synchronization output signal, a digital oscilloscope is used with a sampling rate of 1 GHz, resulting in a resolution of one nanosecond. The scope was triggered on the falling edge of the signal and observed the following falling edge one millisecond later. The triggered signal was recorded over 3,500 times and the results were analyzed with a user-written program. The results show, that the signal

has a maximum of 12 and a mean deviation of 5.8 nanoseconds (fig. 5). It is assumed that the two peeks come from the control loop within the netX.

The measurements show that the synchronization output signal is very accurate and can be used in very precise tasks.

As described above, the estimated delay to invoke a callback function (Δt_{app}) has a too high a jitter to be used in IRT systems. To verify this estimation, a digital oscilloscope was used to measure the time delay between the synchronization output signal and the output of the application. Both mark the event by producing a falling edge on the output. This time the trigger was put on the falling edge of the synchronization output signal and the following falling edge of the output from the application was observed. The triggered signal with a sample rate of 10 MHz, and a resolution of 100 nanoseconds, was recorded about 18,000 times. Again a user-written program was used to analyze the recorded data and to calculate the different delays. The resultant distribution of the delay between the synchronization output signal and the application output is shown in fig. 6.

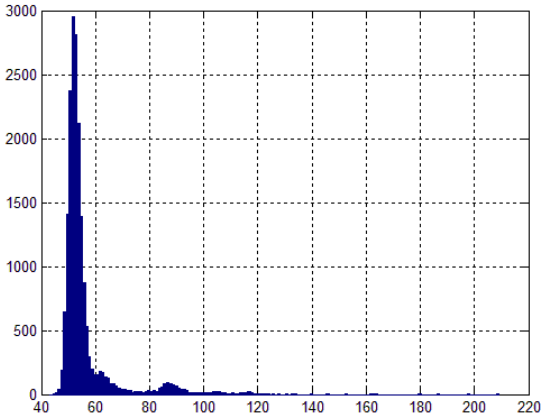

Figure 6. Disitribution of Δt_{app} in µs

The mean delay has a value of 56 microseconds and the maximum measured delay is 208 microseconds. As the distribution shows only 0.16% of the recorded delays is larger than 120 microseconds, however it still has to be considered.

The measurement delivers an estimation of the accuracy of cycle in Profinet devices when using RTC3 transmission. Furthermore it gives an idea about the minimum delay in the hardware latch to eliminate the jitter in the application. Calculation of the jitter for the latch itself is described in clause 4.1of this paper.

3.3. Measured accuracy of the distributed clock in a Profinet system

The cycle time represents one property of a Profinet device, but the deviation of the synchronization between the different devices is equally important.

With the measurement it is aimed to show, the possibilities that Profinet offers to synchronize different devices with each other. Within the measurement setup there is a Profinet network with three devices, transmitting their data as RTC3 every millisecond. As in the measurements before, for the controller a CPU315-F2 PN/DP from Siemens, and for the IO devices the netX development kits from Hilscher are used. The most important information about synchronization is the synchronization output signal of each network controller. Each device must have a hardware latch between the Input/Output and the application to eliminate the jitter of the callback function (see above). The jitter of the synchronization signal and the jitter in the added hardware latch determine the possible accuracy.

In order to guarantee that the synchronization mechanism of Profinet works properly, the following network topology is formed (fig.7). The figure also shows the calculated delay on the cables between each device. A signal needs approximately 5 nanosecond to cover a distance of one meter, as it travels with 2/3 of the speed of light. In this example the calculated delay on the cable between the Device2 and Device3 is 400 nanoseconds, that is 80 meters times 5 nanoseconds per meter.

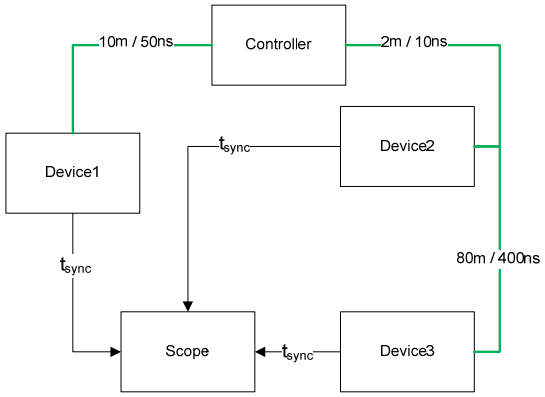

Figure 7. Measurement Setup

In order to verify the accuracy of the synchronization, the synchronization output signal (t_{sync}) of the three devices was measured simultaneously with a sampling rate of 10 GHz, hence a resolution of 0.1 nanosecond. The Oscilloscope is triggered on the edge of the Device1 and shows every edge nearby. With a user-written program all the edges of the 2,000 recorded images are detected and the difference (δt_{sync}) between them was calculated.

The distribution of δt_{sync} between device 2 and 3 is illustrated in figure 8. The represented normal distribution has a mean of -7.4 and a standard deviation of 6.8 nanoseconds. The measured jitter equals to 45.4 nanoseconds. When using hardware latches, these distribution parameters nearly act as a substitute for the possible accuracy in Profinet networks.

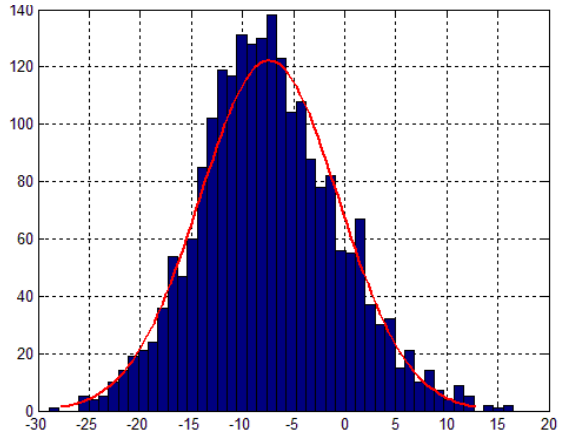

Figure 8. Distribution of δt_{sync} (spatial) in ns

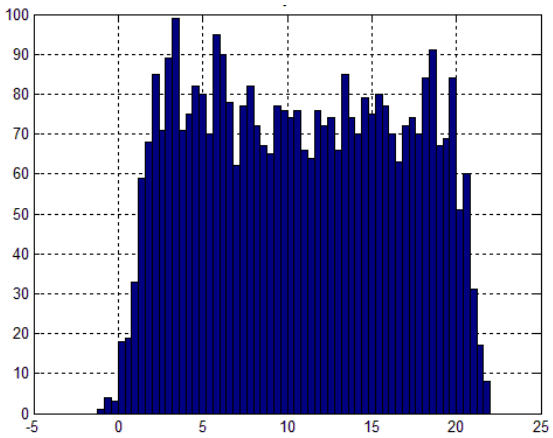

Figure 9. Distribution of δt_{io} (temporal) in ns

4. Improved architectures

As the last section in this paper an evaluation of the accuracy of the proposed system with the hardware latches is provided.

4.1. Accuracy of the improved architecture

If the clocks in the hardware latch work asynchronous with the network, a uniform distributed delay occurs in the latch. The minimum delay in the latch equals to zero and the maximum delay is the cycle time in the hardware latch. In the following simulation examples the engaged clock in the hardware latches operates at 100 megahertz, hence the uniform distributed delay has a minimum of 0 and a maximum of 10 nanoseconds. As the parameters of the distribution are known, the simulation model can be done easily.

In the first simulation example the relative temporal consistency [5] of the improved architecture is evaluated. The precision is defined by the difference of cycle time of the synchronization output signal (δt_{sync}) and the difference in the hardware latch (δt_{delay}). Given that this difference (δt_{sync}), is already measured and added to the generated uniform distributed array, which represent δt_{delay}. This is carried out in order to estimate the resultant total jitter of the output. In figure 9 the resultant distribution is illustrated. When simulated a minimum deviation of -0.93, a maximum of 21.93 and a mean value of 10.8 nanoseconds were obtained. Thus the estimated jitter for such a device has the value of 22.86 nanoseconds.

In the second example the delay between output signals of two different Profinet devices (fig. 10), is evaluated. The two devices are synchronized with each other, but they have a normal distributed difference in-between them (δt_{sync}), as was measured in the previous section.

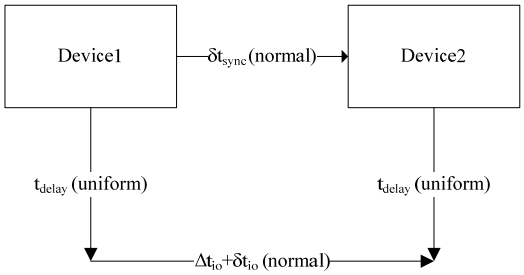

Figure 10. Schematic overview

Since the two latches in the example work with the same t_{delay}, Δt_{io} equals to zero and only δt_{io} has to be considered. To simulate the output difference between the two devices, the result of δt_{sync} is taken from the measurement and the difference of two generated uniform distributed arrays with the same size is added. The arrays simulate the difference of the two different hardware latches (δt_{delay}).

The result is again normally distributed with nearly the same mean of -7.3 and a standard deviation of 7.7 nanoseconds (fig. 11). The maximum simulated jitter is 48.7, but can rise in worst case scenario to 65.4 nanoseconds.

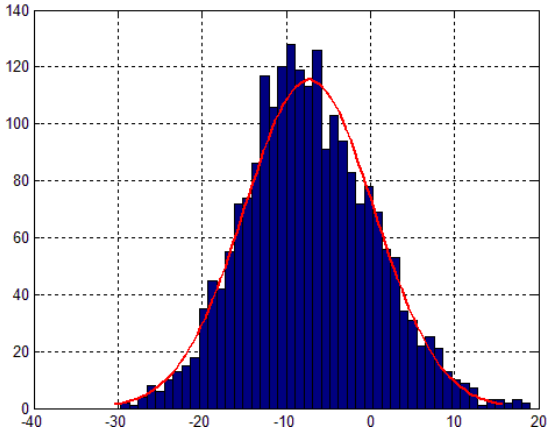

Figure 11. Distribution of δt_{io} (spatial) in ns

The distributions for the two examples represent on one hand the possible accuracy of a device itself and on the other hand the accuracy between different devices in a Profinet network. Since the mechanism stays the same, the results apply equally to synchronized inputs and to system with more than two devices.

The change of the delays in the different hardware latches doesn't affect the deviation.

4.2. Accuracy of the "turning arrow"

Knowing the possible accuracy of a Profinet network using IRT for exchanging the cyclic data, the possible precision of the "turning arrow" system can be calculated(figure 1). The following formulas show the correlation of the delay between the input and the output of two devices in a Profinet network (Δt_{io}) and the corresponding angle of the illuminated arrow on the disc (α_{disc}).

$$\alpha_{disc} = \Delta t_{io} * \omega_{disc}$$

If the delay is constant, the arrow seems to stand on a predefined position (α_{disc} is constant as well) and it is given by the delay and the speed of the disc. But since the delay (δt_{io}) has a normal distributed deviation, the position will have the same deviation. In this example the distributed deviation and the maximum jitter of the target angle is calculated when the disc is turning at 12,000 RPM. The calculations are based on the previous presented result of simulations and measurements. The aim is to control the arrow so that it stands still at zero degree. Hence, the delay between the input and the output device should be zero as well. In figure 10 the result of the simulated delay between two outputs of different devices with the improved architecture is shown. In fact the result with an output and an impulse input stays exactly the same. When the results of the simulation with the angular velocity of the disc are multiplied, the distributed deviation of the target angle is obtained (fig. 12).

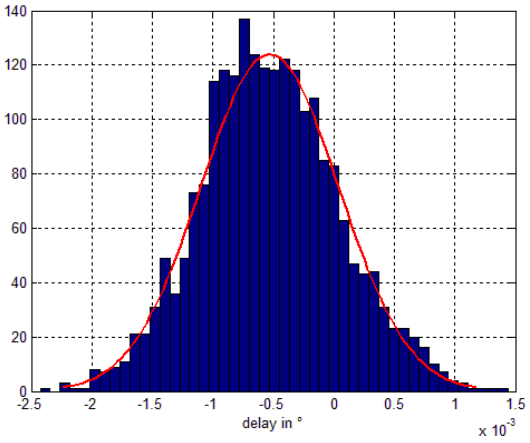

Figure 12. Deviation of the target angle in °

The results show a mean of -0.00053° and a standard deviation of 0.00057°. Hence the arrow stands in the mean -0.0005° of the target position. The simulated arrow moves within 0.0039°.

The same deviation occurs, when the delay between the input and the output is changed and so is the angle of the illuminated arrow.

The maximum deviation of the arrow depends on the speed of the disc (see formulas above). In fig. 13 the calculated maximum jitter of the arrow with a disc speed from 0 to 20,000 rpm is shown. For the value of the maximum jitter between the output and the input of the two devices, the result from previous simulation (fig. 11) is taken (65.4ns).

It can be seen from the results that even if the disc turns at 20,000 RPM the maximum possible jitter of the illuminated arrow is still only 0.008°.

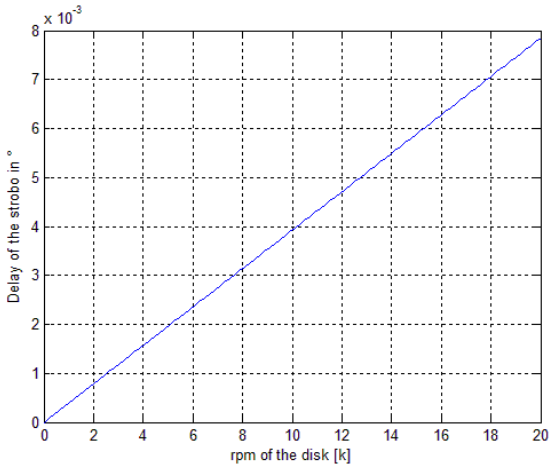

Figure 13. Jitter of arrow, different disc speed

5. Summary and Conclusion

The proposed improvement gives the users the opportunity to realize Profinet IO-devices with a relative temporal consistency within 20 nanoseconds and a spatial consistency within 65 nanoseconds. Unlike the standard, with the proposed approach any datum can be chosen to read or write inputs and outputs. At the moment the user has to implement the transfer mechanism of the demanded datum of an input or an output at the application level. The Profinet standard delivers the possibilities to put the system into practice. If a fixed timing of the IO is needed, the user can avail of the existing mechanisms of Profinet.

To verify the simulated data in a real Profinet network, the proposed architectures and the presented "turning-arrow" system will be realized in the near future.

References

[1] IEC 61158: Industrial communication networks - Fieldbus specifications, a set of standards published by International Electrotechnical Commission and available at www.iec.ch

[2] IEC 61784-1: Industrial communication networks - Profiles - Part 1: Fieldbus profiles, International Electrotechnical Commission, Edition 3.0 2010-07, ISBN 978-2-88912-807-5

[3] IEC 61784-2: Industrial communication networks - Profiles - Part 2: Additional fieldbus profiles for real-time networks based on ISO/IEC 8802-3, International Electrotechnical Commission, Edition 2.0 2010-07, ISBN 978-2-88912-808-2

[4] M. Felser, "Real-time Ethernet - industry prospective", Proceedings of the IEEE, Vol. 93, Issue 6, pp:1118 – 1129, June 2005

[5] J. D. Decotignie, "Ethernet-Based Real-Time and Industrial Communication", Proc. Of the IEEE, VOL. 93, June 2005

[6] M. Felser, "Fieldbus based isochronous automation application", Proc. of 14th IEEE ETFA2009, Sept. 2009

[7] G. Cena, I.C. Bertolotti, S. Scanzio, A. Valenzano, C. Zunino, "Evaluation of EtherCAT Distributed Clock Performance", Industrial Informatics, IEEE Transactions on page(s): 20 - 29, Volume: 8 Issue: 1, Feb. 2012

[8] G. Cena, I.C. Bertolotti, S. Scanzio, A. Valenzano, C. Zunino, "Evaluation of EtherCAT Distributed Clock Performance," Industrial Informatics, IEEE Transactions on , vol.8, no.1, pp.20-29, Feb. 2012

[9] M. Schumacher, J. Jasperneite, K. Weber, „A new Approach for Increasing the Performance of the Industrial Ethernet System PROFINET",Factory Communication Systems, 2008. WFCS 2008. IEEE International Workshop on page(s): 159-167, May 2008

[10] P. Ferrari, A. Flammini, D. Marioli, A. Taroni, F. Venturini "Experimental Analysis to Estimate Jitter in PROFINET IO Class 1 Networks", Proc. of 11th IEEE ETFA2006, Sept. 2006, Prague CZ, CF-002151

[11] P. Ferrari, A. Flammini, D. Mariolo, A. Taroni, F. Venturini, "Evaluation of timing characteristics of a prototype system based on PROFINET IO RT_Class 3", Proc. of 12th IEEE ETFA2007, University of Brescia – DEA

[12] Siemens, ERTEC 400, Manual and Datasheet, available at http://support.automation.siemens.com

[13] Hilscher GmbH, netX500, Protocol API, PROFINET IO RT IRT Device, Revision: 8 (V3.4.x.x), Available: http://www.hilscher.com

[14] J. Jasperneite, K. Shehab, K. Weber, "Enhancements to the time synchronization standard IEEE-1588 for a system of cascaded bridges" Factory Communication Systems, 2004. Proc. 2004 IEEE International Workshop on page(s): 239- 244, 22-24 Sept. 2004

[15] Hilscher GmbH, netANALYZER NXANL 50-RE, Datasheet, available at http://www.hilscher.com