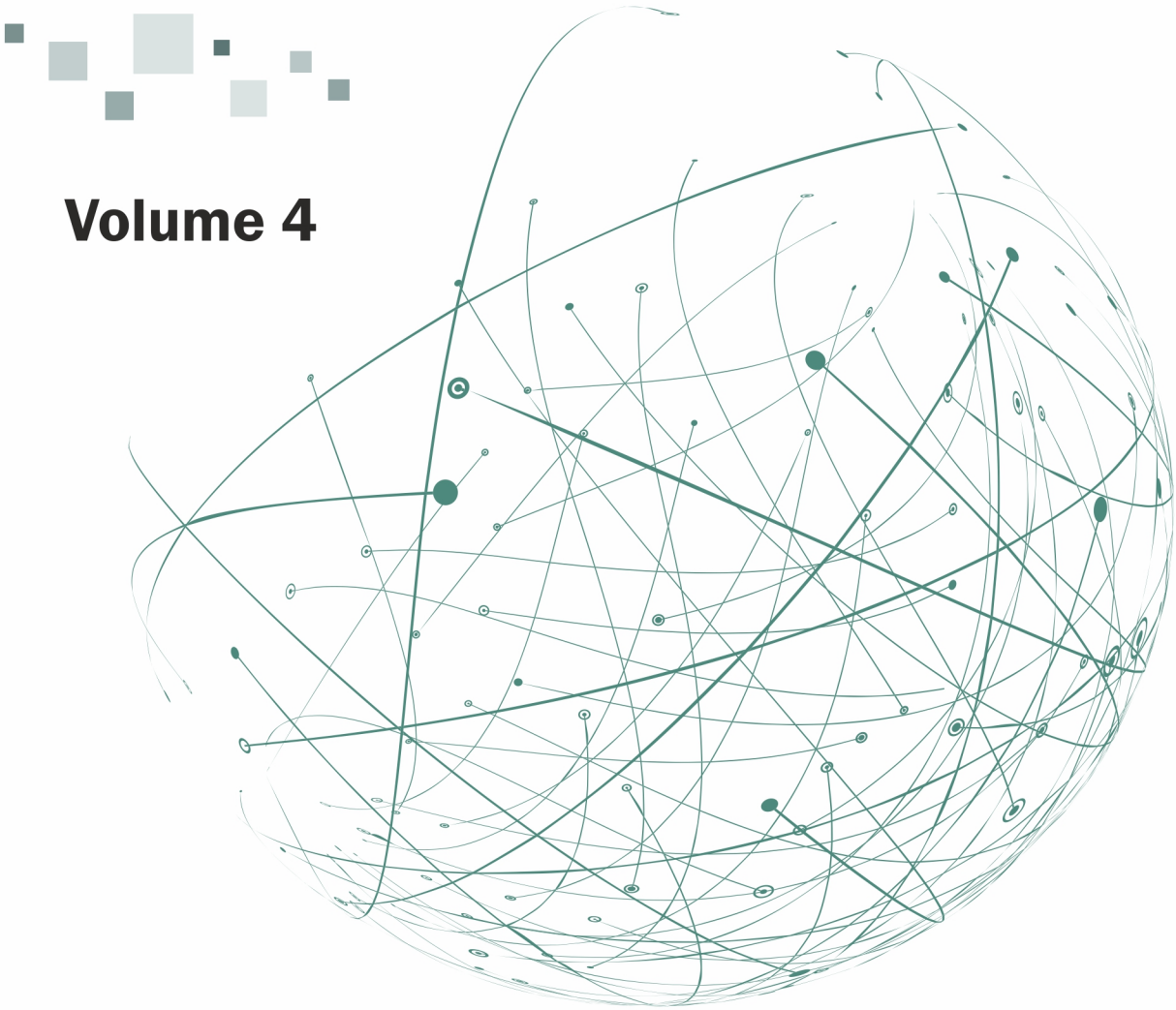
Sergey Y. Yurish
Editor

Advances in Networks, Security and Communications

Volume 4



Advances in Networks, Security and Communications

Volume 4

Sergey Y. Yurish
Editor

Advances in Networks, Security and Communications

Volume 4

International Frequency Sensor Association Publishing

Sergey Y. Yurish
Editor

Advances in Networks, Security and Communications
Volume 4

E-mail (for orders and customer service enquires): ifsa.books@sensorsportal.com

Visit our Home Page on http://www.sensorsportal.com

ISBN: 978-84-09-82029-0

BN-20251230-XX

BIC: TJK

Acknowledgments

As Editor I would like to express my undying gratitude to all authors, editorial staff, reviewers and others who actively participated in this book. We want also to express our gratitude to all their families, friends and colleagues for their help and understanding.

Contents

Contributors

Fabrice Belvèze
Gipsa-lab, Université Grenoble Alpes, CNRS, Saint Martin d'Hères, 38400, France

J. Börcsök
University of Kassel, Dept. Computer Architecture and System Programming, Kassel, Germany

Jean-Marc Brossier
Gipsa-lab, Université Grenoble Alpes, CNRS, Saint Martin d'Hères, 38400, France

Miqueu Paul
Gipsa-lab, Université Grenoble Alpes, CNRS, Saint Martin d'Hères, 38400, France

Laurent Ros
Gipsa-lab, Université Grenoble Alpes, CNRS, Saint Martin d'Hères, 38400, France

M. H. Schwarz
University of Kassel, Dept. Computer Architecture and System Programming, Kassel, Germany

Takayoshi Yokota
Faculty of Information Design, Tokyo Information Design Professional University, 2-7-1 Komatsugawa, Edogawa-ku, Tokyo, 132-0034, Japan

Preface

The accelerating integration of networks, cybersecurity mechanisms, and communication systems is redefining the principles by which contemporary platforms are engineered, verified, and brought into operation—from industrial automation and intelligent transportation to high-data-rate digital links and sensor-rich cyber-physical infrastructure. *Advances in Networks, Security and Communications, Volume 4* arrives at a moment when "connectivity" is no longer a single engineering objective, but a negotiated balance among reliability, latency, functional safety, resilience to interference, and exposure to adversarial behavior. The chapters in this volume reflect that reality: they do not treat networks, sensing, and communication theory as isolated disciplines, but as interacting layers whose coupling determines real-world performance and trustworthiness.

What makes this Book Series timely is its emphasis on practical rigor. Each contribution addresses a concrete class of operational constraints—non-deterministic wireless channels in safety contexts, GNSS limitations and vulnerabilities in vehicle positioning, and frequency-selective distortion in baseband communications—then builds toward methods that are implementable, analyzable, and testable. Together, the chapters illustrate a shared theme: "robustness" is not a single technique but a system property achieved through careful modeling, measurable metrics, and disciplined algorithm design.

Chapter 1 focuses on one of the most consequential questions in contemporary industrial and safety-critical networking: how to reason about—and engineer—communication channels whose behavior is not fully certified or fully predictable. Building from the established "black channel" and "white channel" concepts used in functional safety communications, the authors motivate a *grey-channel* approach—one that remains "uncertified" in the strictest sense, yet is *observable* and *measurable* in ways that allow the system to adapt and maintain required safety integrity. The chapter develops protocol structures (FIDES frames) that can activate or deactivate safety features based on monitored channel conditions, such as bit errors, losses, and latency changes. By combining redundancy strategies, error detection (notably CRC choices with specific Hamming distances), sequence and timing mechanisms, and structured message formats, the work illustrates how safety-oriented communication can become adaptive rather than static—an increasingly relevant perspective for wireless and interference-prone environments.

Chapter 2 shifts from protocol design to mobility intelligence, addressing vehicle localization under conditions where satellite navigation can be degraded, denied, or attacked. GNSS has long been the default positioning backbone, but dense urban environments, multipath, tunnels, and intentional interference increasingly expose its limitations. The chapter presents a method for estimating vehicle position using onboard MEMS sensors—accelerometers, gyroscopes, and geomagnetic sensors—by correlating sensor signatures against reference runs.

Beyond the algorithmic details, the chapter's broader message is about *redundancy of perception*: localization can be treated as a fusion of physical motion cues and map structure, offering resilience when traditional radio-based positioning becomes unreliable. For practitioners in intelligent transportation systems, autonomy support layers, or safety-critical fleet operations, the techniques described here point toward pragmatic alternatives and complements to GNSS-centric architectures.

Chapter 3 returns to core communication engineering, tackling the persistent challenge of high-data-rate baseband transmission over frequency-selective channels. Such channels introduce intersymbol interference (ISI) that can undermine detection performance unless the receiver uses equalization strategies capable of handling channel memory. The chapter surveys and develops equalization methods grounded in maximum likelihood sequence estimation (MLSE) implemented via the Viterbi algorithm, and then explores how *channel shortening* techniques can reduce effective channel memory to make optimal or near-optimal detection computationally feasible. This chapter will resonate strongly with readers who build or analyze high-throughput digital links—whether in wired access, wireless baseband, industrial communication, or embedded communications systems—where the tension between performance and computational complexity is central. It reinforces an essential lesson: receiver design is an optimization problem over accuracy, latency, and implementability, not a single "best" algorithm.

Volume 4 of *Advances in Networks, Security and Communications* demonstrates that engineering progress is often made not by chasing novelty alone, but by *tightening the loop between observability and design*: measure what matters, model it carefully, and build mechanisms that remain valid under imperfect conditions. Whether the reader's focus is safety-adaptive protocol behavior, sensor-driven localization beyond GNSS, or equalization strategies that unlock high-rate performance in selective channels, the chapters share a common spirit of pragmatic depth.

This book will be valuable to multiple communities precisely because it addresses the *interfaces* between disciplines: researchers, graduate students, industrial automation and safety engineers. It is our hope that this volume will serve both as a reference for current practice and as a catalyst for further work—work that treats networks not merely as conduits for data, but as accountable components of systems that must remain dependable, secure, and efficient in the real world.

Dr. Sergey Y. Yurish

Editor
IFSA Publishing *Barcelona, Spain*

Chapter 1
Flexible Communication Protocols – Steps Towards a Grey-Channel Approach

M. H. Schwarz and J. Börcsök

1.1. Introduction

Communication protocols play a vital issue in many industrial areas. It started with fieldbus communication such as *Modbus*® [1] in 1979 for the automation area, followed by *CAN*® [1, 2] for the automotive sector and others such as *Profibus*® [1, 2], *Interbus*® [1, 2] and many others. In the late 1990's, early 2000's, many manufacturers adopted their communication protocols to Ethernet based technology, such as *Profinet*® [2], *EthernetPowerlink*® [1] or *EtherCAT*® [1] and many others. In parallel, in 1996, the first *OPC*® [3-5] communication (*OLE for Process Control*) was established, which is based on a client server architecture that transfers current process data, named *OPC-DA*® (*Data Access*) [3-5] and is not based on any particular manufacturer's protocol. This was necessary as end-users were using different protocols and it was difficult to get all information together in e.g. a control room [3-5], also at this time, not all protocols were fully open and standardized. These objective facts resulted in many problems and fault-prone communication, therefore *OPC*® was very successful and several other communication specifications and extensions were developed such as *OPC-AE*® (*Alarm and Events*) [3-5], to provide additional information if an alarm occurred or *OPC-HDA*® (*Historical Data Access*) [3-5], to provide not only the current process data, but also previous data and to aggregate them to determine trends and irregularities. The OPC foundation recognized that to establish several client-server communications to access actual data, historical data and alarms or events makes the system complex also security issues that were not considered in 1996 as a problem to be solved directly on the client/server level. Several additional extensions dealing with security, or redundant server and client strategies were developed to provide fault tolerant systems. The *OPC-UA*® (*Unified Architecture*) [3-5] specification was released in 2006, where all different UPC specification were united and also security and fault tolerant aspects were considered right from the beginning. Also, at this time, companies started to develop or extend their existing protocols to safety related protocols such as *OpenSafety*® [6],

M. H. Schwarz
University of Kassel, Dept. Computer Architecture and System Programming, Kassel, Germany

ProfiSafe® [7] or *Fail Safety over EtherCAT*® (FSoE) [8]. Very often certification bodies were right from the beginning involved. In 2018, the OPC foundation released the *OPC-UA TSN*® specification [9] that allows the use in time-critical applications and finally, a safety specification was released, in order to deal with safety critical data. The industrial organization HMS-Networks [10] publishes frequently the market shares of the different fieldbuses. Fig. 1.1 shows that wireless has a portion of 7 % and states a current market growth of 8 %. Table 1.1 shows the markets shares and growths from 2015, 2019 and 2022 [10]. While in 2015 wireless communication was not relevant, it occupied a small portion with a large market growth of 30 % in 2019. Manufacturers realized the innovation and potential of wireless communication and users started to replace wired communication with wireless in order to reduce the cabling [10].

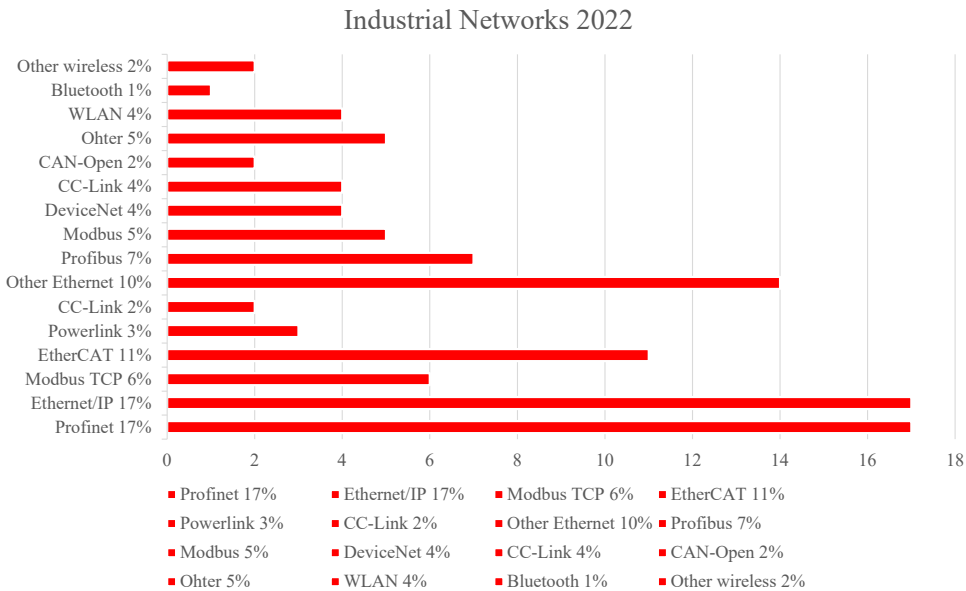


Fig. 1.1. Market shares of industrial networks [10].

The market growth of 8 % in 2022 is due to fact that Ethernet based communication has grown stronger and customizers waiting for products of 5G and its impact. Functional safety and cyber security are big issues in wireless communication that have to be constantly improved.

Table 1.1. Market shares of industrial networks in different years [10].

	2015 Market		**2019 Market**		**2022 Market**	
	Share	***Growth***	***Share***	***Growth***	***Share***	***Growth***
Fieldbus	66 %	7 %	35 %	-5 %	27 %	4 %
Ethernet	34 %	17 %	59 %	20 %	66 %	10 %
Wireless	-	-	6 %	30 %	7 %	8 %

Wireless communication is getting popular in automation [11], in process control [12], robotics [13], logistic [14] and many other industrial areas. Although, products on LTE and 5G are awaited, research, investigations and specifications on 6G are in full progress [15]. Nevertheless, issues about safety and security are present and valid [16]: then wireless communication is non-deterministic and more interference-prone as wired communication [17]. Furthermore, safety issues and security subjects have to be combined as sources of faults [18-20]. New methods arise to determine if an intrusion is present or only noise. Intruders can be identified by their radio frequency fingerprint [21, 22]. If the fingerprints are known, then the transmitter can be classified as trusted otherwise as an intruder. The additional advantage is that the detection is carried out at low level of the communication and the data is rejected at low level and not withdrawn at high level, where the data is already in the process of the receiver [21, 22]. Fig. 1.2 presents different scenarios that can arise during transmission [23, 24]. The first scenario shows a normal transition without any disturbances. Scenario two shows that the transmission is blocked by an obstacle. This is not possible with wired communication. Also, it cannot be stated if the obstacle was placed unintentionally (can be stated as a safety issue) or on purpose (can be stated as a security issue). The third scenario shows a communication with interferences. Disturbances can also happen in wired communication when for example the wires are badly shielded. However, disturbances can occur more often in wireless communication and can also happen intentionally (as a security issue). Scenario four shows an intrusion which is much easier in wireless communication. Traditionally, fieldbuses are not connected to the internet and are fully isolated.

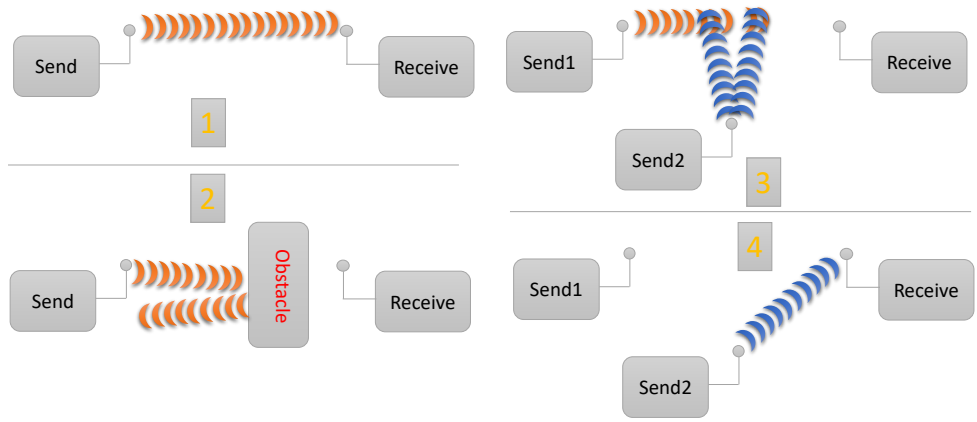


Fig. 1.2. Different communication scenarios [23, 24].

Each component provides a certain failure rate towards an overall failure rate [25]. This overall rate is used to classify the system and its integrity. As the communication is operating continuously the *Probability of Failure per Hour* (PFH) is used to determine the actual *Safety Integrity Level* (SIL). Table 1.2 presents the PFH values and its corresponding SIL [25].

Table 1.2. Relation of PFH and SIL.

SIL	SIL / PFH	
	PFH for safety system	***PFH for communication channel***
4	$< 10^{-8}$	$< 10^{-10}$
3	$< 10^{-7}$	$< 10^{-9}$
2	$< 10^{-6}$	$< 10^{-8}$
1	$< 10^{-5}$	$< 10^{-7}$

This value defines the possibility that an error occurs but is not detected and this leads to a safety critical situation [25]. Wired communication may contribute 1 % of the safety value of the overall system [26]. Enough methods exist to detect errors and to decrease the PFH value. In communication especially in wireless communication a temporal increase of errors due to disturbances can occur. Fig. 1.3 illustrates the behavior of CRCs in combination with a changing *Bit Error Ratio* (BER). The first CRC (blue) is No. 13 with a *Hamming Distance* (HD) of 6, while the second CRC (red) is No.0 with a *Hamming Distance* of 3, both listed in Table 1.6, respectively. The *PFH* value increases if the *BER* increases as well. For a *BER* of 10^{-4} the blue *CRC* would be out of scope of any SIL value the red CRC can manage it quite well.

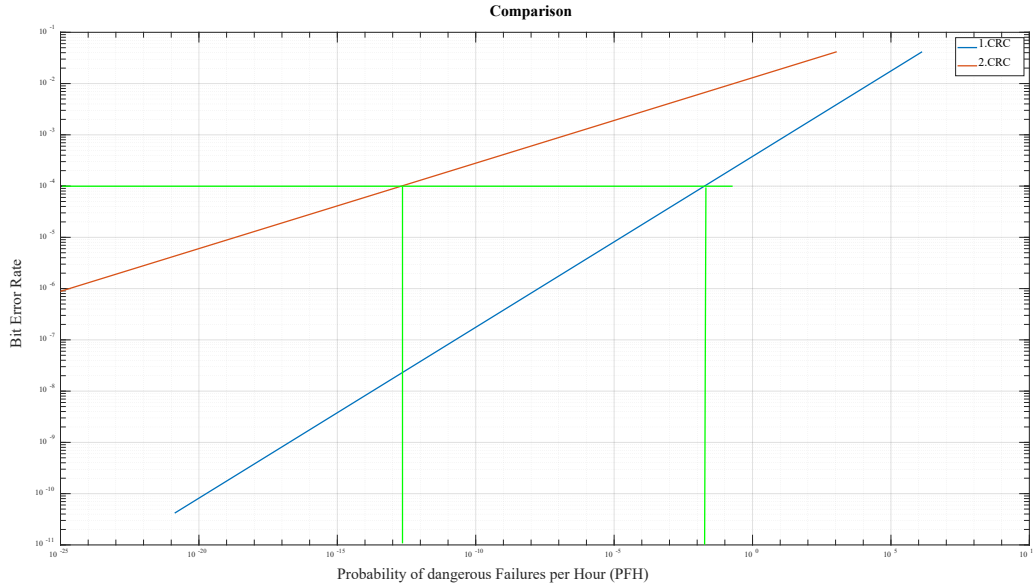


Fig. 1.3. PFH behavior for two different CRCs.

This illustrates that due to a changing *BER* the *PFH* can change as well and finally the *SIL* value. This means the system is operating in a not allowed operating mode. Therefore, to continuously monitor the *BER* and as a result the *PFH* can be beneficial in order to detect, if occasionally, temporally or permanently the defined SIL level is violated.

The remaining chapter is structured as follows: Section 1.2 is introducing the different channel approaches and discusses why it is beneficial to introduce a *Grey Channel*. Section 1.3 illustrates the different communication models and discusses the selected method. In the following section different error detection methods are presented and discussed. The equations to calculate the *PFH* values are indicated in Section 1.5 followed by the introduction of the new protocols in Section 1.6. Section 1.7 exposes the results and validation. Finally, Section 1.8 concludes and states future work.

1.2. Black, White and Grey Channel

In safety communication related to automation and process industries, two different approaches exist to describe the communication channel between different devices. The first approach is the white channel. The standard IEC 61783 part 3 [27], defines a white channel es follows:

"3.1.1.44 white channel: communication channel in which all relevant hardware and software components are designed, implemented and validated according to IEC 61508"

This implies, that all elements of the communication have to be examined for certification as the remaining structure. The result is that two strictly separated communication strands exist one for safety communication and one for non-safety communication. An alternative is the black channel approach [27]:

"3.1.1.3 black channel: communication channel without available evidence of design or validation according to IEC 61508"

Fig. 1.4 explains the black channel method where all safety demands are executed in the safety layer. The black channel is defined as everything below the safety layer. It does not matter, if there already exist any protection or error detection method, it assumes that no method exists and every protection is carried out only in the safety layer.

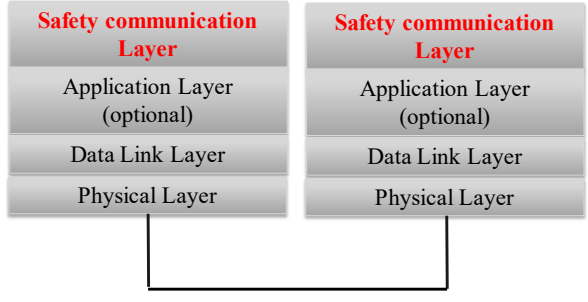


Fig. 1.4. Black Channel method.

The safety layer possesses all relevant safety methods which are explained and detailed in the next sections. The advantage of the black channel approach is that standard, non-safe

communication can be mixed with safety communication. The safety protocol can be embedded in the standard protocol [23, 24]. This is often performed in automation were safe and non-safe messages are sent in one frame [23, 24]. In an information header, it has to be indicated if the data is for a safe or non-safe communication. Fig. 1.5 illustrates the combination of different message types in one transmission line. Fig. 1.6 [23, 24, 27] sketches a more detailed picture how both communications safe and non-safe can operate together and how strictly separated the non-safe and safety application has to be, so that they do not affect with each other.

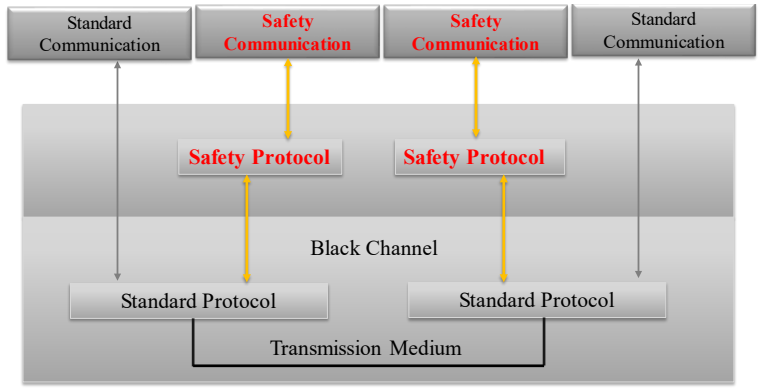


Fig. 1.5. Combination of safe and non-safe protocols [23, 24, 27].

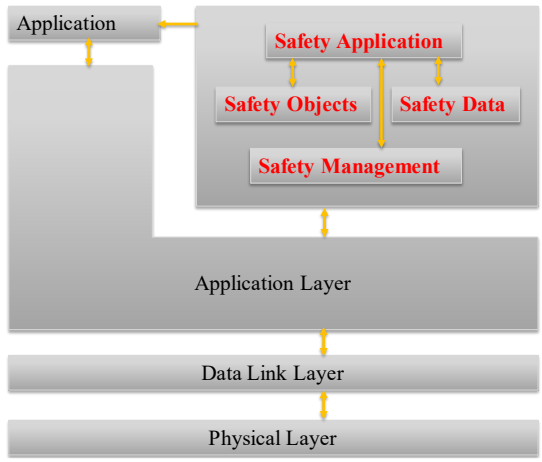


Fig. 1.6. Combination of safe and non-safe application [23, 24, 27].

The term grey channel is originally defined for railway systems in the DIN EN 50159 [28, 29] and has an identical meaning as the black channel in the Standard IEC 61783-3 [27].

In the work of Åkerberg [30] a grey channel is defined as a non-safe communication channel lacking a black channel assessment. Here, safe related data is transmitted over a grey channel for information purposes only [24, 30].

In Åkerberg et al. [31] an interesting definition of a grey channel is presented, where a black channel has not to fulfil any requirements. A grey channel is a communication which fulfils requirements or possesses some constraints, which are at least a minimum bandwidth or maximum amount of latency. The authors argue that nowadays a grey channel would be a better approach for safety communication [24, 30].

In this article the communication channel should be observable and at least the number of data bit errors and data bit losses can be detected and counted. It is not necessary to estimate the *Bit Error Ratio* (BER) or the *Bit Less Ratio* (BLR), as errors and losses are counted. It is beneficial to measure the transition time of a sent request and received answer in order to determine the latency and to determine if the latency starts to increase. Therefore, a grey channel can be defined as follows:

Grey Channel: communication channel without available evidence of design or validation according to IEC 61508, but is observable and bit errors and bit losses can be detected and counted and chances in the latency can be detected as well.

1.3. Communication Models

The standard IEC 61783 part 3 [27], defines four different communication models related to safety communication. The four models present different fault detection methods and show possible implementation constructions; other models exist and can also be used. The implementation can be done in hardware or software.

1.3.1. Model A – Two Safety Layers One Message

This model as shown in Fig. 1.7 possesses two safety communication layers that perform independently the safety procedures and verify it. If the cross-checking reveals a difference, then reasonable actions have to be taken in order to maintain safety. The safety communication layers are not as independent as in Model D.

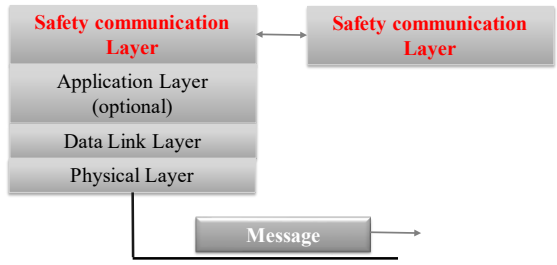


Fig. 1.7. Model A [27].

1.3.2. Model B – Redundant Layers, Messages and Transmission

This model possesses every part twice. It consists of two safety layers; two communication lines and two separate messages are generated. The message can be identical or inverted; also, different transmission mediums can be used (Fig. 1.8).

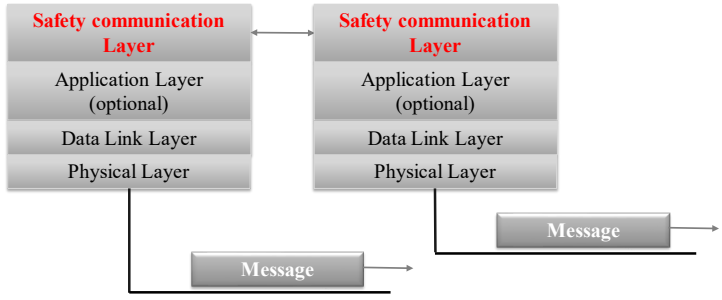


Fig. 1.8. Model B [27].

1.3.3. Model C – Redundant Layers, Messages, One Transmission

This model is similar to Model B. The system is entirely redundant besides it uses the same transmission line, not a separate as Model B (Fig. 1.9).

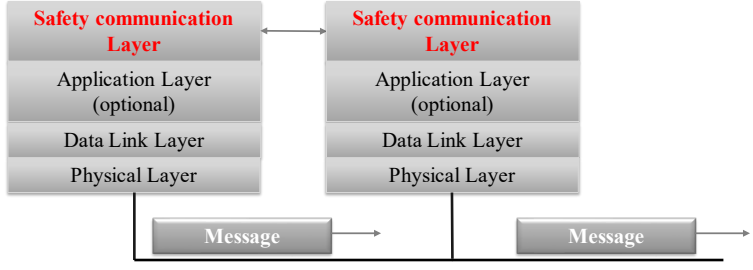


Fig. 1.9. Model C [27].

1.3.4. Model D – Two Safety Layers and Two Messages

This model can be seen as a mixture of Model A and C. It generates two messages and consists of two separate layers. As a variant of Model D, it is also possible to generate only one message (Fig. 1.10).

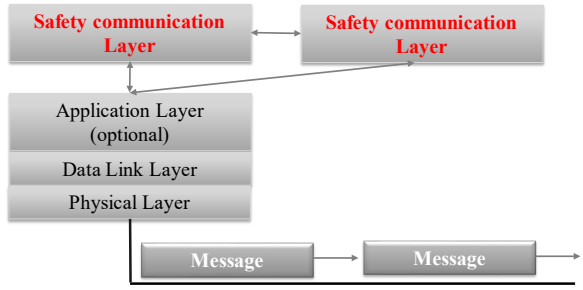


Fig. 1.10. Model D [27].

1.4. Error Detections Methods

Different communication standards [27, 28] and reports [17, 23, 24] on communication protocols describe different classes of errors in messages and provide error detection mechanisms to spot them for example for *ProfiSafe*® [2, 7, 32, 33] and *OpenSafety*® [6, 34, 35]. Data bits in messages can change due to some technical errors, intentional or unintentional interference, noise and other faults. In general, the following identified errors have to be detected. The numbers and letters after the errors and detection method are used in Table 1.3 to state which error can be detected using which method:

- *Unintended repetition* (1): A receiver gets an outdated message [27, 24];
- *Corruption* (2): Data bits are falsified during transmission [27, 24];
- *Incorrect sequence* (3): messages reached the receiver not in the order as sent [27, 24];
- *Loss* (4): A sent message is not reaching the receiver or the acknowledgment of a received message has not arrived at the original transmitter [27, 24];
- *Unacceptable delay* (5): The defined maximum time to receive a message is exceeded [27, 24]. Additional, also a minimum time can be defined;
- *Insertion* (6): A message is received which belongs to an unexpected or unknown source [27, 24]. The message cannot be rated as correct, unintended repetition or incorrect sequence [27];
- *Masquerade* (7): a non-safe relevant message is classified as a safety message [27, 24].

Errors in a message can be detected with the following procedures:

- *Sequence number* (A): A number is integrated in the message that is incremented from message to message [27, 24];
- *Time stamp* (B): The absolute or relative time is added to the message. It is necessary that transmitter and receiver are synchronized [27, 24];
- *Time expectation* (C): The message has to reach the message within a predefined time. It can be the time between two received message or the time between a transmitted and received message [27, 24]. A synchronization is not necessary;

- *Connection authentication* (D): It is an exclusive source and/or destination identifier in the message that describes the logical address of the safety relevant transmitter [27, 24];
- *Feedback message* (E): The receiver confirms the correct arrival of the message [27, 24]. The acknowledge message can consists of a short message that only acknowledges the received message, it can additionally consist of the original message or additional information that verifies the message is received correctly;
- *Data integrity assurance* (F): Redundant data is added to the message is such a way that it can be detected if the data got falsified. [27, 24]. Normally, a CRC is used;
- *Redundancy with cross checking* (G): The message is sent twice, within one message or as a separate message. The data can be identical or changed e.g. inverted [27, 24].

Table 1.3. Error versus detection method.

Method/Error	***A***	***B***	***C***	***D***	***E***	***F***	***G***
1	X	X					X
2						X	X
3	X	X					X
4	X	X			X		X
5		X	X				
6	X	X		X	X		X
7				X	X		

From Table 1.3 it can also be seen that several methods are needed to detect all classes of errors. Therefore, several methods have to be selected.

1.5. PFH – Equations

The communication standard IEC 61784-3 [25] defined the *PFH* value for communication as follows:

$$PFH = 3600 \cdot v \cdot (m-1) \cdot 100 \cdot P_{ue}\,, \tag{1.1}$$

where v is the number of safety relevant messages per second, m is the number of communication nodes and P_{ue} is the residual probability of error. The value *100* specifies that the transmissions only contributes 1 % to the error rate, as discussed in Table 1.2. To calculate the P_{ue} value, it can be founded on a *Binary Symmetric Channel* (*BSC*) approach [26]. In this approach that a bit changes from value 0 to value 1 or vice versa is equal to probability ε [26, 36].

$$p(1|0) = p(0|1) = \varepsilon \tag{1.2}$$

The equation of P_{ue} can be found in IEC 61784-3 [25] as follows:

$$P_{ue} \leq \sum_{i=d}^{n} \binom{n}{i} \cdot \varepsilon^{i} \cdot (1-\varepsilon)^{n-i}, \tag{1.3}$$

where n is the message length, and d is the *Hamming Distance*. If n is small then the P_{ue} can be calculated exactly. However, often is n large and the computational effort is immense. Therefore, an approach is used by Wacker et al. [37] that derive an upper bound for P_{ue} avoiding to calculate the weights:

$$P_{ue} \leq \frac{72}{121} \cdot \frac{\sqrt{2 \cdot \pi \cdot n}}{2^r \cdot d!} \cdot n^d \cdot \varepsilon^d + R_n(\varepsilon), \tag{1.4}$$

where r is the length of the checksum and $R_n(\varepsilon)$ is the remainder term, and defined as [37]:

$$R_n(\varepsilon) \leq \begin{cases} 2^n \cdot \left(\sqrt{\varepsilon}\right)^n, if\ n \geq 3\ and\ even \\ 2^n \cdot \left(\sqrt{\varepsilon}\right)^{n-1}, if\ n \geq 4\ and\ odd \end{cases} \tag{1.5}$$

When inserting equation (1.4) in equation (1.1), and using the first term of equation (1.5) as remainder, then this results in the following equation:

$$PFH = 3600 \cdot v \cdot (m-1) \cdot 100 \cdot \left(\frac{72}{121} \cdot \frac{\sqrt{2 \cdot \pi \cdot n}}{2^r \cdot d!} \cdot n^d \cdot \varepsilon^d + 2^n \cdot \left(\sqrt{\varepsilon}\right)^n\right) \tag{1.6}$$

In Wacker et al. [37], it is argued when calculating n_{max} that the remainder is small and can be neglected. In equation (1.6) the remainder is used but only for an even data length. Most messages are even when using the adaptive protocols in the next section and where the message has an odd length the difference is very small but makes the computation easier. When it is distinguished between bit error ε and bit loss φ then the following equation can be used [26]:

$$PFH = 3600 \cdot v \cdot (m-1) \cdot 100 \cdot (1-\varphi)^n \cdot \left(\frac{72}{121} \cdot \frac{\sqrt{2 \cdot \pi \cdot n}}{2^r \cdot d!} \cdot n^d \cdot \varepsilon^d + 2^n \cdot \left(\sqrt{\varepsilon}\right)^n\right) \tag{1.7}$$

The authors in [38] argue that all bit losses are detected by the receiver as the data frame is getting violated. Therefore, it is paradox but true, that the *PFH* value in this equation is actually getting better if more bits getting lost. The variables from equation (1.6) and (1.7) can be more or less changed. Variable m, number of safety nodes, is mostly fixed. Variable v is the number of safety relevant messages per second. V varies depending if the messages have to be sent more than once, depending of the protocol or if for example the method redundant messages is active or not. Variable r (bits of CRC) and d (Hamming Distance) depend on the selected CRC. Variable n, is the length of data k and length of CRC r, as shown in equation (1.8). The variables bit error ratio ε and bit loss ratio φ are counted by the algorithm.

$$n\left(message_{length}\right) = k\left(Data_{length}\right) + r\left(CRC_{length}\right) \tag{1.8}$$

However, for the next section equation (1.6) is used as all bits that are not correct, does not matter if lost or false are getting counted as errors. When bits are getting lost or added this can be described as follows:

$$Error_{Loss/Added} = \frac{\sum_0^n |Expected\ Bits - Received\ Bits|}{Expected\ Bits} \tag{1.9}$$

If data gets falsified then this can be calculated as shown:

$$Error_{Original/Inverted} = \frac{\sum_0^n Falsified\ Bits}{Expected\ Bits} \tag{1.10}$$

Inserting equation (1.9) and (1.10) into equation (1.7) results in:

$$PFH = 3600 \cdot v \cdot (m-1) \cdot 100 \cdot (1-\varphi)^n \cdot \cdot \left(\begin{array}{l} \frac{72}{121} \cdot \frac{\sqrt{2 \cdot \pi \cdot n}}{2^r \cdot d!} \cdot n^d \cdot \cdot \left(Error_{\frac{Loss}{Add}} + Error_{Ori._{\overline{Invert}}} \right)^d + \\ +2^n \cdot \left(\sqrt{\left(Error_{Loss/Add} + Error_{Ori./Invert} \right)} \right)^n \end{array} \right) \tag{1.11}$$

The reason, why the errors get differentiated is that two different functions are verifying if the bit is falsified or if the number of received bits is correct. The final equation is shown below:

$$FH = 3600 \cdot v \cdot (m-1) \cdot 100 \cdot (1-\varphi)^n \cdot \left(\begin{array}{l} \frac{72}{121} \cdot \frac{\sqrt{2 \cdot \pi \cdot n}}{2^r \cdot d!} \cdot n^d \cdot \left(\frac{\sum_0^n |Expected\ Bits - Received\ Bits|}{Expected\ Bits} + \frac{\sum_0^n Falsified\ Bits}{Expected\ Bits} \right)^d + \\ +2^n \cdot \left(\sqrt{\left(\frac{\sum_0^n |Expected\ Bits - Received\ Bits|}{Expected\ Bits} + \frac{\sum_0^n Falsified\ Bits}{Expected\ Bits} \right)} \right)^n \end{array} \right) \tag{1.12}$$

1.6. Adaptive Protocols

This section describes all developed protocols that varies the behavior depending on the calculated PFH value. Those protocol structures called FIDES (Finding Errors), to give them a name.

1.6.1. General Structure

Normally, in wired communication, large data packets are sent, as typically the error rate is low and the wires should be appropriately shielded [2]. One example is the category of summation frame of a field-bus system. Data for different field-bus nodes are chained up and send along the transmission line. Fig. 1.11 shows a schematic, where separate slots are used for specific nodes or processes. The data field can contain process data or commands or both depending on the actual protocol or configuration. Also, each data slot for a particular communication node, can also have its own data CRC protection.

Fig. 1.12 shows another example similar to Modbus [1, 2, 23] where each client or communication node is addressed followed by a command and data. Several client or node register can be read at once if they are adjacent or data can be written to several nodes, also, if they are next to each other. The message is protected by a CRC.

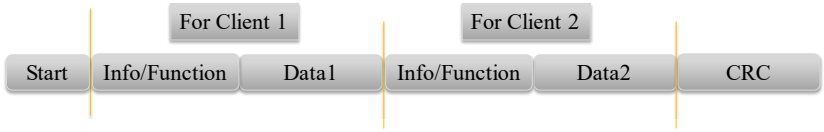


Fig. 1.11. Similar to a Sensor-Actuator Fieldbus [1, 2, 23].

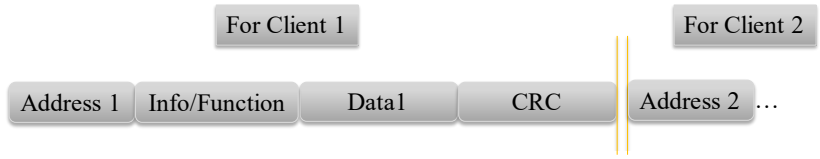


Fig. 1.12. Modbus Fieldbus [1, 2, 23].

In wireless communication disturbances can occur more frequently therefore, it is beneficial to keep messages short, as shown in Fig. 1.13. CRC can detect if errors are in the message but cannot correct them. So, if one bit is wrong the entire message has to be resent, if there is enough time.

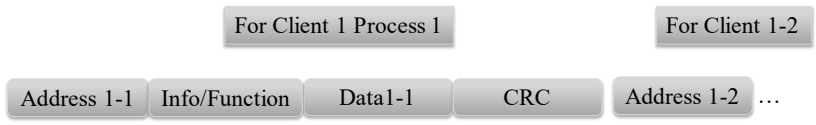


Fig. 1.13. Wireless Communication [23].

In the proposed protocols the message length is kept very short, so even if data belongs to the same communication node it is put in a separate message. Therefore, if a node controls several automation or control loops, they are independently accessed. If a message for a particular loop gets disrupted this does not interfere with the other loops. The general protocol structure is shown in Fig. 1.14:

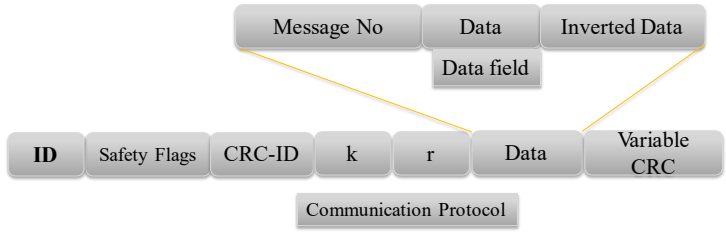


Fig. 1.14. Protocol structure [23].

Table 1.4 explains the additional information / header of the protocol. With the *identifier* (ID) a particular loop or client can be identified. As mostly underlaying protocols are used

as typical for a black channel approach, this identifier is not replacing the addressing part but can indicates more precisely, which safety loop or process is meant. The safety flags indicate which safety method is activated. As the safety flags varies in the protocols, they are explained in detail in the subsections. *K* defines the data length, in Fides 1 up to 1012 bits can be sent, therefore, 10bits are necessary. In Fides 2 only up to 100 bits are used, depending on the selected CRC. Therefore, 7bits are necessary. *R* defines the CRC length at the end of the message. Data and CRC vary on the amount of data to be sent and on the number of bits that are added for protection at the end of message depending of the selected CRC.

Table 1.4. Protocol bits.

	Bitlength	***Comments***
ID	4	Client/Process identifier
Safety Flags	4	Switches safety methods on/off
CRC-ID	4	Selected CRC
k	10/7	Data length
r	4	CRC length
Data	Variable	Data + add. Information
CRC	variable	CRC protection

1.6.2. Fides 1

Fides 1 is the first structure that was developed and studied. It used Model A, as detailed in Section 1.3. In Table 1.5, the different safety flags are presented. With the first bit it is indicated if the *time stamp* is added to message. The second bit specifies if the method *consecutive number* is active, followed by the CRC, which can also be switched off or on and finally if the data is sent inverted and added to the message as shown in Fig. 1.14.

Table 1.5. Fides 1 Safety bits.

Bit position	***Bitlength***	***Comments***
1	1	Time stamp
2	1	Consecutive number
3	1	CRC active
4	1	Redundant, inverted data

The different CRCs that can be selected are shown in Table 1.6. It represents the first group; Fides 1 is only operating with these CRCs. The CRCs are used from Koopman [39]. The composed group has CRCs with *Hamming Distances* from 3 to 6, have a CRC length from 6 bits to 16 bits and a data length from 57 bits up to 1012 bits.

Table 1.6. CRC Group 1.

No.	***CRC***	***r***	***k***	***d***
0	0x33	6	57	3
1	0x65	7	120	3
2	0xe7	8	247	3
3	0x119	9	502	3
4	0x327	10	1013	3
5	0x5b	7	56	4
6	0x83	8	119	4
7	0x17d	9	246	4
8	0x247	10	501	4
9	0x583	11	1012	4
10	0xbae	12	53	5
11	0x212d	14	113	5
12	0xac9a	16	241	5
13	0x372b	14	57	6
14	0x573a	15	114	6
15	0x9eb2	16	135	6

Fig. 1.15 shows the data-skeleton without any protection method, and is the minimal set to be transmitted [23, 24]. This structure has no protection at all, but different protection methods can be added.

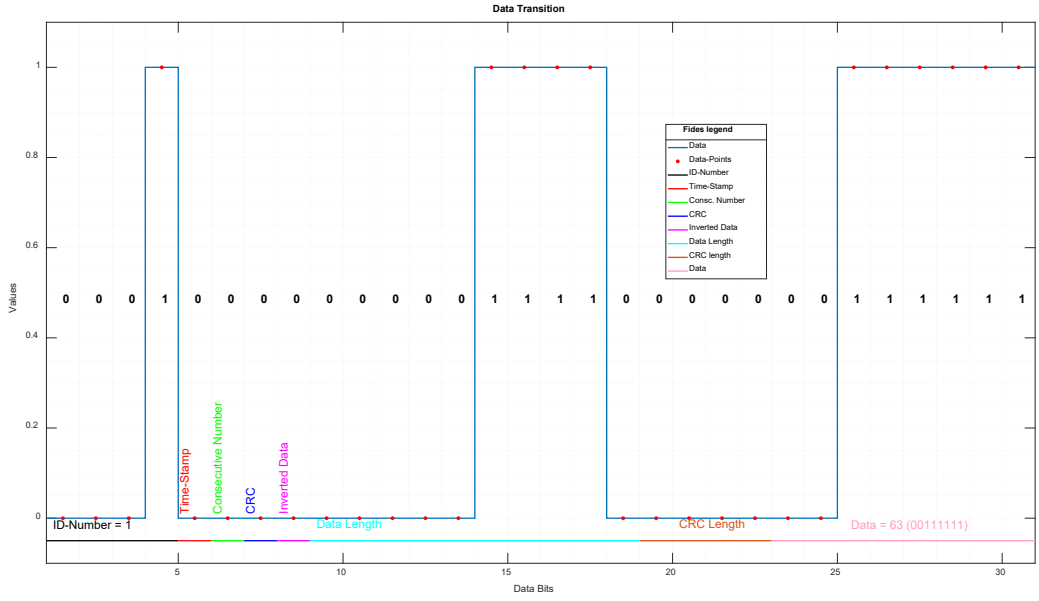


Fig. 1.15. Minimal structure without any protection.

In the beginning the consecutive number was put before the CRC-ID. This position was not ideal, as the position of the CRC-ID shifted depending if the feature was switched on or off. Fig. 1.16 shows the old position with consecutive numbers and Fig. 1.17 demonstrate the new position with three successive messages.

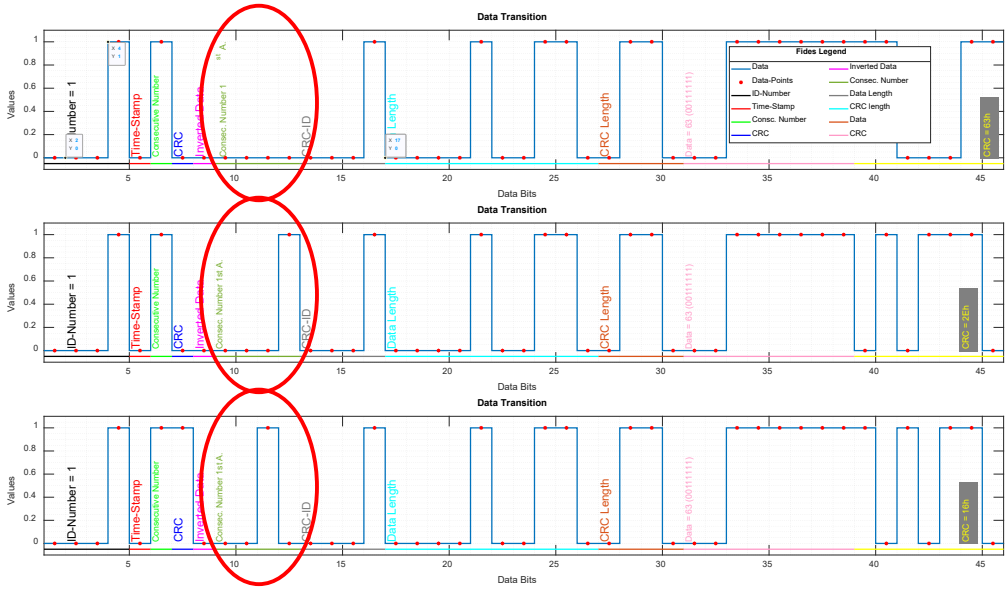


Fig. 1.16. Consecutive number at front.

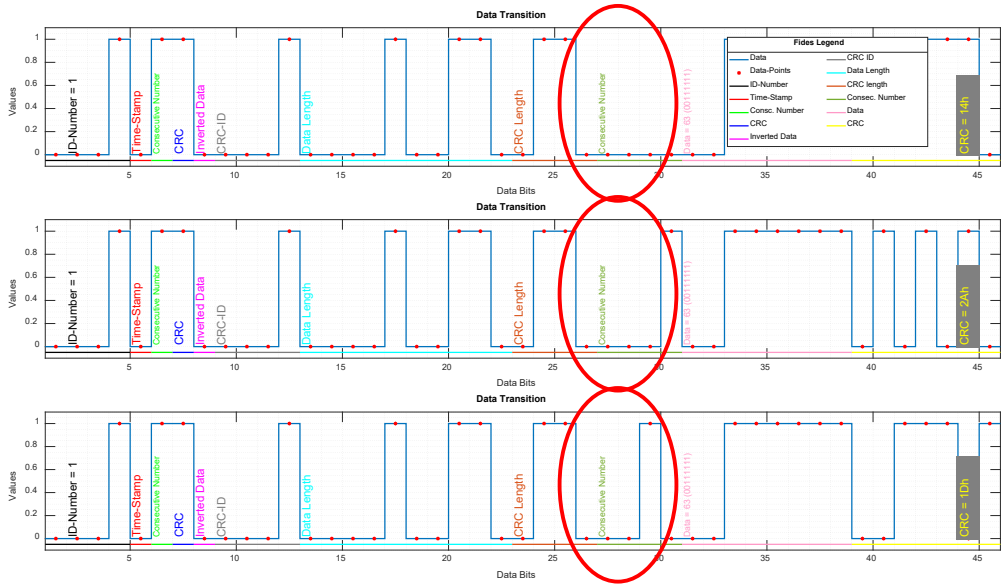


Fig. 1.17. Consecutive number in data field.

Fides 1 inverts only the data section if activated and not the entire message as a part of an additional safety method. The data is better secured. The inversion is demonstrated by equation (1.13). The first line is the original data, which gets falsified in the second line. When using an XOR function with the inverted data, then the number of falsified data bits can be identified. This is particular interesting when counting the number of errors in a message [23, 24].

$$\begin{array}{cccccccc} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \quad \textit{Original data} \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \quad \textit{Faulty data} \\ & & & & \textit{XOR} & & & \\ \underline{0} & \underline{1} & \underline{0} & \underline{1} & \underline{0} & \underline{1} & \underline{0} & \underline{1} \quad \underline{\textit{Inverted data}} \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \quad \textit{Result} \end{array} \tag{1.13}$$

1.6.3. Fides 2

The second approach is Fides 2, which uses a combination of Model A and Model D, as described in Section 1.3. Also, it takes the finding of Fides 1 into account. The safety flags got a different meaning and switches new features as described the Table 1.7.

Table 1.7. Fides 2 Safety bits.

Bit position	***Bitlength***	***Comments***
1	1	Full redundancy
2	1	Message number
3	1	CRC group 1 or group 2 active
4	1	Redundant, inverted data

It was decided that the feature CRC cannot be switched off anymore. Although, inverted redundant data can be used to detect errors and also to count them, but it is difficult to argue that a message is not sent with a CRC protection. Fides 2 actually uses two CRC tables, the first group as shown in Table 1.6 and used by Fides 1 and a new group as shown in Table 1.8. Those CRC's have a larger *r* but also a higher *Hamming distance d*. For the length of *r,* 4 bits does not fit, therefore, the minimal value of 21 is subtracted so that 4 bits are enough. The receiver has to add the value 21 in order to get the correct length. This has to be done only if the CRC group2 is used. The consecutive message number was also kept, as it is a simple but important feature to detect errors such as [23, 24]:

- Repetition of a message;
- Loss of a message;
- Insertion of a message;
- Wrong sequence.

Fig. 1.18 displays three messages with the features: *consecutive number* with regular data and the inverted transmitted data in one message. If the data is corrupted then a comparison of data and inverted data with the help of an XOR function can be performed [23, 24]. The new feature full redundancy [34, 23, 24] inverts the entire message and send it as a separate message. The only values that are not inverted are the ID, the bit inverted

message and the CRC. The ID is not inverted as the client has to know for whom the message is, the bit *full redundancy* indicates that this is the redundant message and CRC should not be inverted so that the normal CRC checking procedure is used. Otherwise, the message has to be inverted first and the CRC is checked afterwards. Fig. 1.19 shows the original message and the inverted message. It also consists of the inverted process data. The redundant message is also important for to count the errors. At the end, the number of bit errors has to be detected in order to calculate the PFH value, as stated in the equations above. A simple XOR function is used to determine the differences in the messages and those are then added together.

Table 1.8. CRC Group 1.

No.	*CRC*	*r*	*k*	*d*
0	0x12faa5	21	106	7
1	0x189efe	22	105	7
2	0x5e2419	23	106	7
3	0x880ee6	24	231	7
4	0x289cfe	22	105	8
5	0x469d7c	23	105	8
6	0xcba785	24	105	8
7	0x4429686	27	48	9
8	0xeaa72ab	28	99	9
9	0x1e150a87	29	100	9
10	0x242c0684	30	100	9
11	0x12b00d4	25	40	10
12	0x32def69	26	40	10
13	0x51aff9a	27	41	10
14	0x1e150a87	29	100	10
15	0242c0684	30	100	10

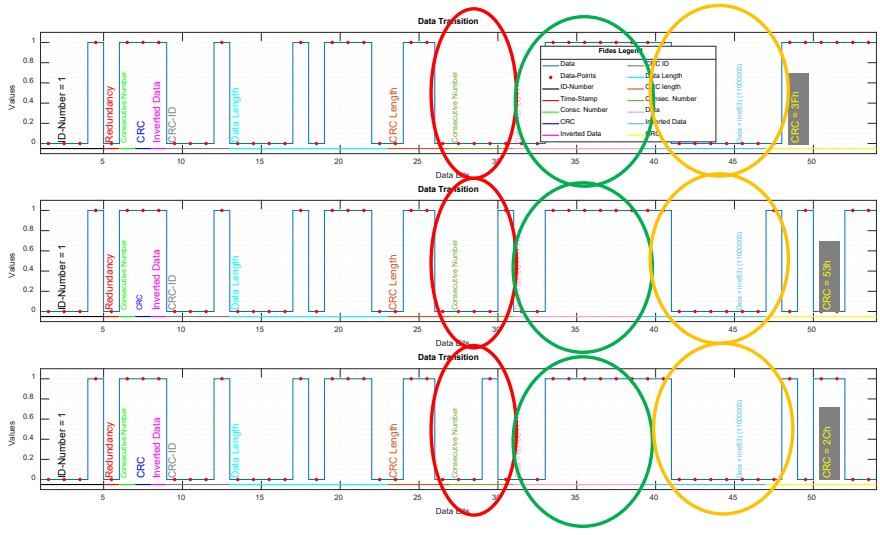


Fig. 1.18. Consecutive number, data and inverted data.

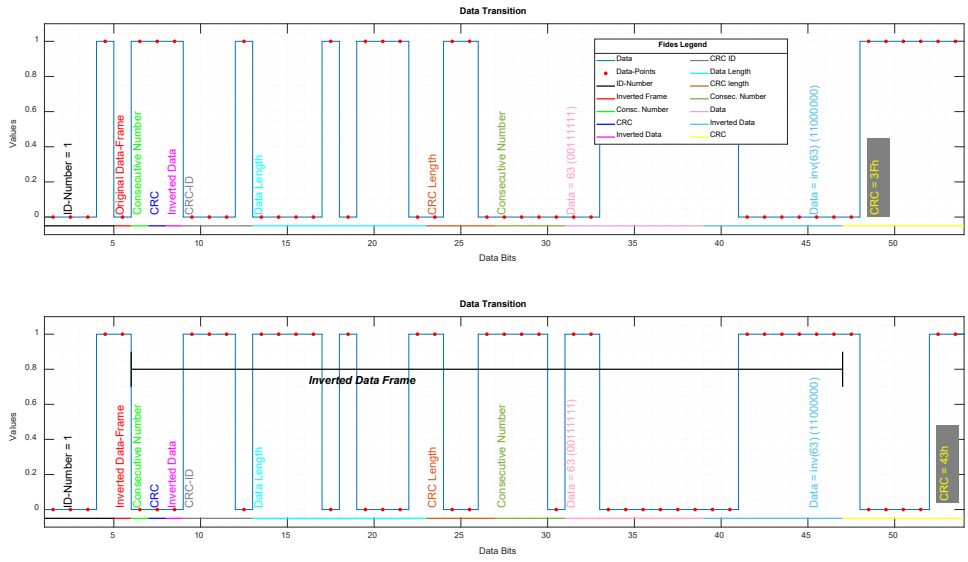


Fig. 1.19. Redundant message.

1.6.4. Fides 3

The third approach is Fides 3 that uses a combination of Model A and Model D, as before. Also, it takes the finding of Fides 1 and Fides 2 into account. Fig. 1.20 shows the new structure, where variable *r* is erased and the message number has now a fixed and permanent position. The place of the safety flag reserved for the consecutive message number is now used to indicate if this current message is the redundant message as shown in Table 1.9. This means, the first safety flag indicates if the method *full redundancy* is active and the second bit states if this message is actually the redundant message.

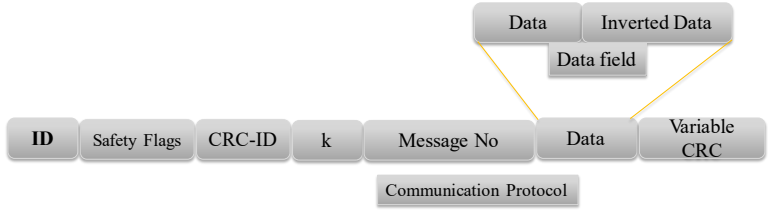


Fig. 1.20. Protocol structure.

Table 1.9. Fides 3 Safety bits.

Bit position	***Bitlength***	***Comments***
1	1	Full redundancy
2	1	Redundant Message
3	1	CRC group 1 or group 2 active
4	1	Redundant, inverted data

Fig. 1.21 shows again the original message and the inverted message. The ID, the method *full redundancy* and the flag redundant message are not inverted. The receiver knows exactly how to treat the message.

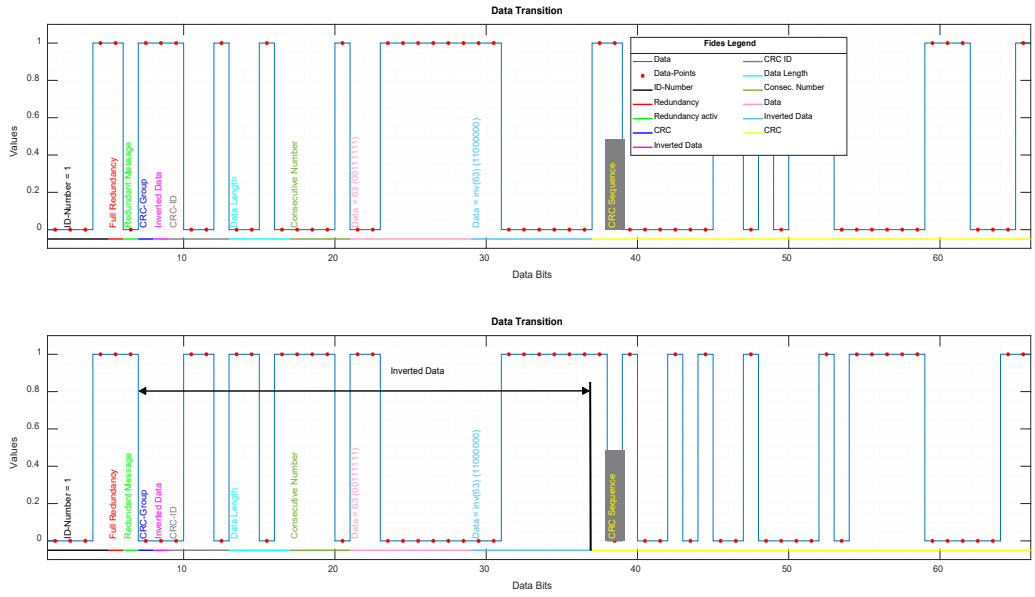


Fig. 1.21. Message and redundant message.

1.6.5. Fides 4

The fourth approach is Fides 4, that uses again a combination of Model A and Model D. Also, it takes the findings of the previous structures into account. When calculating the PFH value, equation (1.12) is used for an even data transmission, that included header, data and CRC. The selected CRC is the only part in the Fides frame that can make the number of bits to be sent odd. In order to calculate a precise PFH value, Fides 4 distinguishes if the number of bits are even or odd. If they are already even, the frame is selected as shown in Fig. 1.20 of Fides 3. If the number of bits is odd. An *Even-bit* is added before the CRC, that makes now the frame even, as shown in Fig. 1.22 and equation (1.12) can now be used, correctly. Fig. 1.23 shows again the data frame with the added *Even-bit*.

1.7. Tests and Validation

This section discusses the different findings of the protocols and describes which one are taking into further considerations and which one are not more considered.

1.7.1. Tests with Fides 1

Fides 1 was the first protocol to investigate the different features. The time stamp is an often-used method to detect if a message is on time or outdated. This method relies on a synchronization of all participants in the communication loop. Easier to establish is the method time expectation. A message has to arrive within a predefined time, as shown in Fig. 1.24.

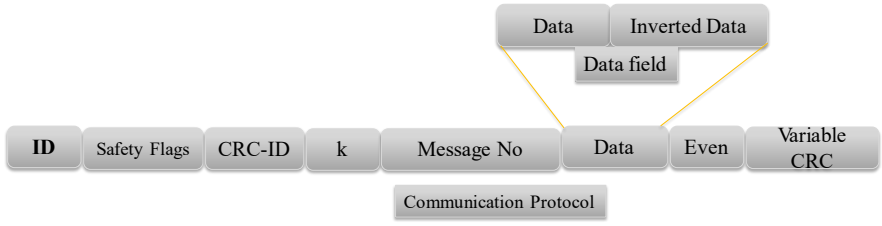


Fig. 1.22. Message with an Even Bit.

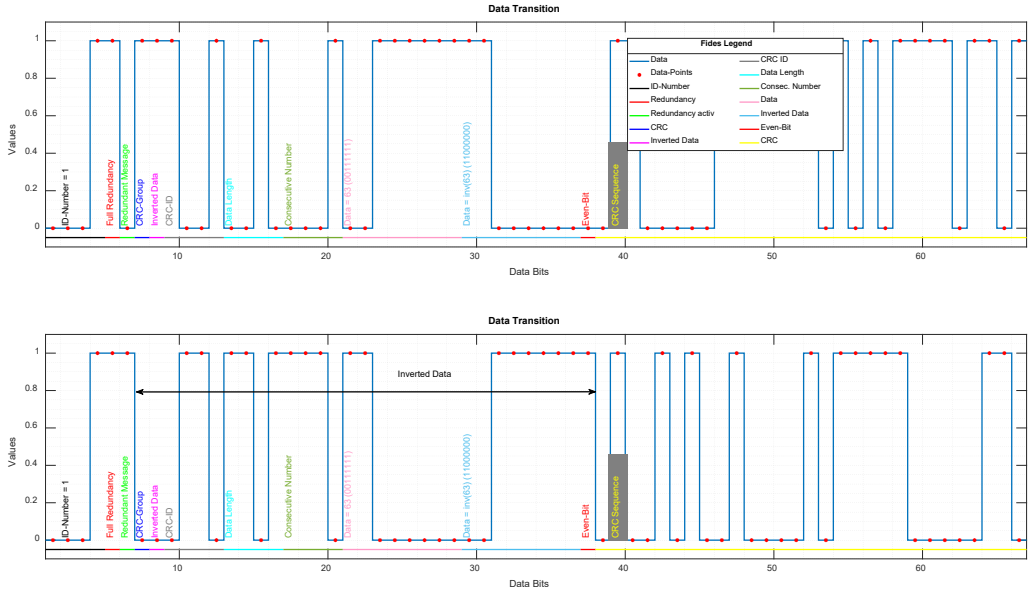


Fig. 1.23. Message and redundant message with Even-bit.

If the message is too late or does not reach the receiver at all, the time elapses and the problem is detected. If a minimal reaction is defined, which is the minimum to receive the message, and it arrives faster before the minimum time elapses then this message should be treated with caution as this could be a message error of the category *insertion*, as described before, or it can be an intrusion. Fides 2 is taken those findings into account.

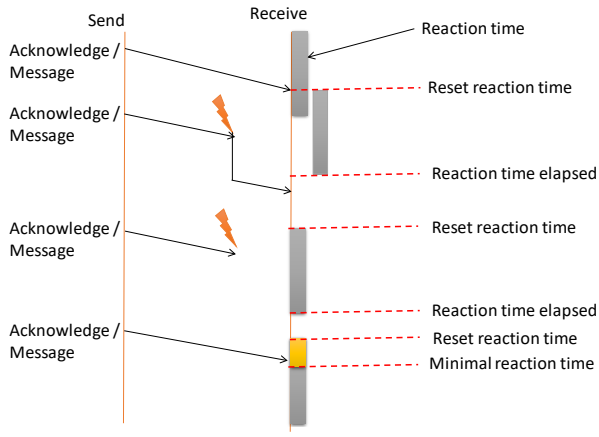


Fig. 1.24. Time expectation.

1.7.2. Tests with Fides 2

Fides 2 worked well, but also revealed some drawbacks. As the sequence number is an important feature, it is mostly active anyway, so therefore, this flag is not necessary. The flag redundancy was used to identify the redundant inverted message. It should have two flags, one indicates the feature redundancy is used, and one that actually tells that this is the redundant message. Therefore, the omitted sequence number flag can be reused as the indication of the redundant message. Fides 2 has also some redundant information as shown in the figure below. The CRC-length is known, as the CRC-ID indicates which CRC should be used.

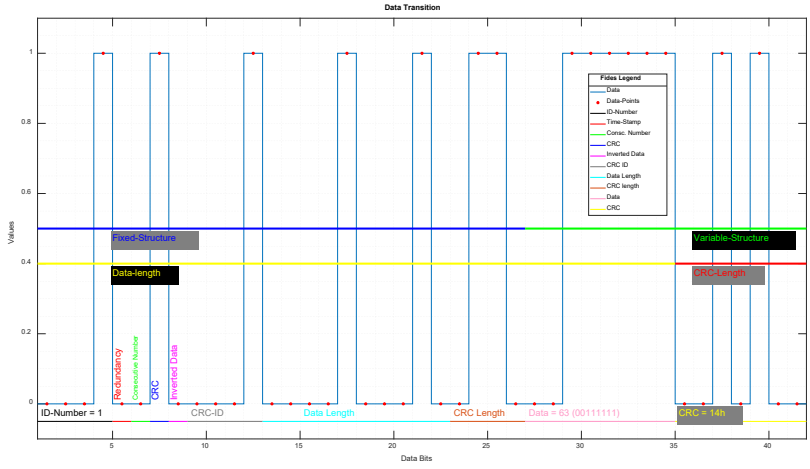


Fig. 1.25. Fixed and variable structure of Fides 2.

Often, data consists of a multiple of bytes and most of the protocol is fixed, the section data length could be shrinking to only the information of the data in bytes that are sent.

1.7.3. Tests with Fides 3

Fides 3 worked well, but showed that when often a CRC was selected that makes the data frame odd and then taking equation (1.12) for PFH for even data frames that when comparing using the equation for odd data frames that an increasing error can be stated. Therefore, it is beneficial to decide beforehand if all frames are either even or odd.

1.7.4. Tests with Fides 4

Fides 4 solved the problem that equation (1.12) can be used without any restrictions. To use the even-bit, if it is placed in the frame as an additional feature: all one-bits were counted and if this number is even then the even-bit is set to zero others one.

1.8. Conclusions and Future Work

Several new data frames for communication are presented that can change its safety features depending on the actual PFH value. The actual decision when a feature should be active or deactivated, was not the main topic, it was more focused which feature is useful and which can be omitted. Fig. 1.26 shows the principle of the observation of the PFH value and therefore for the SIL. The different Fides-frames are a continues investigation of different features and the calculation of the PFH value. Methods like time stamp was omitted and others like expectation time or full redundancy added to the frames. The method *Even-bit* made the PFH calculation more exact as the correct remainder for the even frame is used. However, further investigations are necessary.

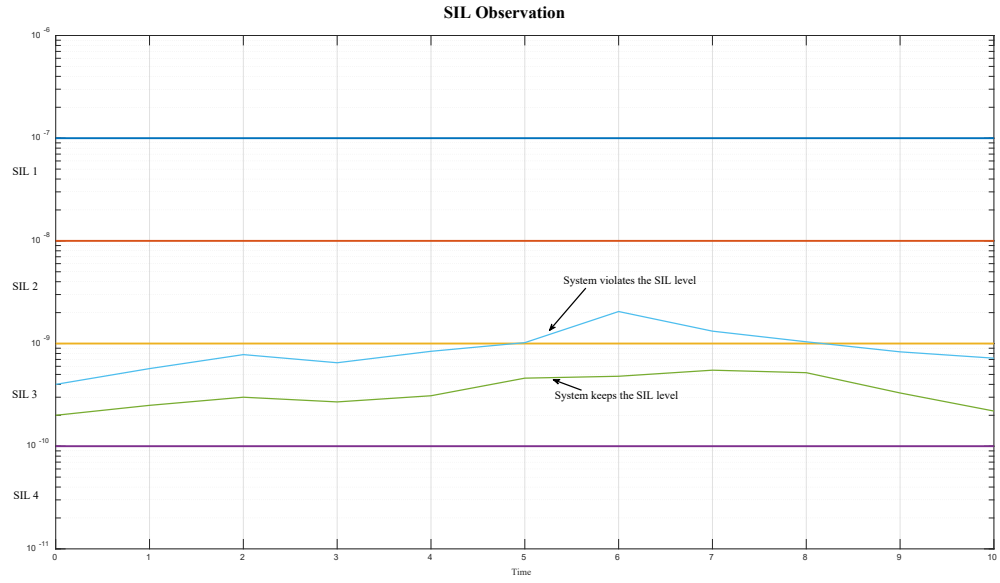


Fig. 1.26. Example of SIL observation [24].

1.8.1. Future Work

The next steps are to investigate to add further bits to the data frame, such as the Even-bit either on fixed positions or on dynamical changing positions, in order to detect, if for example the header structure is violated. To use two or even more CRCs to protect certain parts of the frame, can be useful. The selected CRCs have a large range of CRC lengths. This should be reinvestigated to find CRCs with a similar length *r*. When the frames are used within other protocols then the frame length should be extended with the stuffing bits to a multiple of 8bits. Therefore, a similar length r could be beneficial in order not to add too many stuffing bits. An investigation of security issues such as encryption should be added to the frames. Finally, an effective decision make has to be developed.

References

[1]. S. S. Kumar, Fieldbus and Networking in Process Automation, Second Edition, *CRC Press,* 2021.

[2]. B. Reißenweber, Feldbussystem zur industriellen Kommunikation 2. Auflage, *Oldenburg Industrieverlag GmbH*, München, 2002 (in German).

[3]. J. Lange, F. Iwanitz, T. J. Burke, OPC: von Data Access bis Unified Architecture, *VDE-Verlag*, 2014.

[4]. W. Mahnke, S. H. Leitner, M. Damm, OPC Unified Architecture, *Springer Verlag*, Berlin, 2009.

[5]. M. H. Schwarz, J. Börcsök, A survey on OPC and OPC-UA: About the standard, developments and investigations, in *Proceedings of the XXIV International Conference on Information, Communication and Automation Technologies (ICAT'13)*, Sarajevo, Bosnia and Herzegovina, 30 October 2013 – 01 November 2013, pp. 1-6.

[6]. OpenSafety, https://www.ethernet-powerlink.org/open-safety

[7]. ProfiSafe, https://www.profibus.com/ technology/profisafe

[8]. FsoE, https://www.ethercat.org/en/safety.html

[9]. D. Bruckner, M.-P. Stănică, R. Blair, et al., An introduction to OPC UA TSN for industrial communication systems, *Proceedings of the IEEE*, Vol. 107, Issue 6, June 2019, pp. 1121-1131.

[10]. HMS, www.hms-networks.com

[11]. B&R sichere Datenübertragung per Funk, https://www.br-automation.com/de-at/ueber-uns/presse/sichere-datenuebertragung-per-funk-24-09-2018

[12]. Schildknecht, https://www.schildknecht.ag/produkte/industrial-wireless/

[13]. D. F. N. Gordon, Y. Wu, X. An, Communication-aware motion control for mobile robot applications, in *Proceedings of the 4th Winter IFSA Conference on Automation, Robotics & Communications or Industry 4.0/5.0 (ARCI'24)*, Innsbruck, Austria, 2024, pp. 327-331.

[14]. Schildknecht, Fahrerloses Transportsysteme, https://www.schildknecht.ag /funkanwendung/ fahrerlose-transportsysteme/

[15]. J. Wu, R. Li, X. An, et al., Toward native artificial intelligence in 6G networks: System design, architectures, and paradigms, *arXiv preprint*, 2021, arXiv:2103.02823.

[16]. W. Ikram, N. F. Thornhill, Wireless communication in process automation: A survey of opportunities, requirements, concerns and challenges, in *Proceedings of the UKACC International Conference on Control*, 2010, pp. 1-6.

[17]. L. Gaus, M. Schwarz, J. Börcsök, A theoretical approach to a safety-based predictive adaptation of wireless communication channel parameters in harsh environments, *Safety in Extreme Environments*, Vol. 2, 2020, pp. 93-101.

[18]. E. Mueller, X.-L. Chen, R. Riedel, Challenges and requirements for the application of Industry 4.0: A special insight with the usage of cyber-physical system, *Chinese Journal of Mechanical Engineering*, Vol. 30, 2017, pp. 1050-1057.
[19]. M. N. O. Sadiku, Y. Wang, S. Cui, et al., Cyber-physical systems: a literature review, *European Scientific Journal*, Vol. 13, Issue 36, December 2017, 52.
[20]. E. Ugljesa, H.-D. Wacker, J. Börcsök, Modelling security aspects in safety environment, in *Proceedings of the 7th International Conference on Electrical and Electronics Engineering (ELECO'11)*, 2011, pp. II-46-II-50.
[21]. A. Kara, Practical implications of the use of Radio Frequency Fingerprinting (RFF) in cybersecurity of smart grids, in *Proceedings of the 4th Winter IFSA Conference on Automation, Robotics & Communications or Industry 4.0/5.0 (ARCI'24)*, Innsbruck, Austria, 2024, pp. 346-348.
[22]. S. R. Rondla, Investigation of using artificial intelligence to detect safety critical intrusion, MD Thesis, *University of Kassel*, Kassel, Germany, 2023.
[23]. M. H. Schwarz, J. Börcsök, A communication observer for monitoring the safety integrity level in real-time, in *Proceedings of the 4th Winter IFSA Conference on Automation, Robotics & Communications or Industry 4.0/5.0 (ARCI'24)*, Innsbruck, Austria, 2024, pp. 243-248.
[24]. M. H. Schwarz, J. Börcsök, Adaptive communication protocols – steps towards a grey-channel approach, *Sensors & Transducers*, Vol. 265, Issue 2, May 2024, pp. 127-138.
[25]. IEC 61508, International Standard 61508 Functional Safety of Electrical/Electronic/ Programmable Electronic Safety Related Systems, *International Electrochemical Commission*, Geneva, Switzerland, 2000.
[26]. P. K. Pendli, Contribution of Modelling and Analysis of Wireless Communication for Safety related Systems with Bluetooth Technology, *Kassel University Press*, Kassel, 2014.
[27]. IEC 61784-3, Ed. 2.0, Industrial Communication Networks, Part 3: Functional Safety Fieldbuses, *International Electrochemical Commission*, Geneva, Switzerland, 2010.
[28]. CENELEC: En 50159-2, Railway Applications – Communication, Signaling and Processing Systems, Part 2: Safety Related Communication in Open Transmission Systems, *Beuth Verlag GmbH*, 2001.
[29]. J. Peleska, O. Schulz, Reliability analysis of safety-related communication architectures, in Computer Safety, Reliability, and Security. SAFECOMP 2010, *Springer*, Berlin, Heidelberg, 2010, pp. 1-14.
[30]. J. Åkerberg, Mobile Interaction with safety critical systems: A feasibility study, MD Thesis, *Mälardalen University*, Västerås, Sweden, 2015.
[31]. J. Åkerberg, M. Gidlund, F. Reichenbach, M. Björkman, Measurements on an industrial wireless HART network supporting PROFIsafe: a case study, in *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation (ETFA'11)*, 2011, pp. 1-8.
[32]. M. Friske, H. Schlingloff, H. Barthel, Verifikation und Test des PROFIsafe-Sicherheitsprofils (Verification and tests of PROFIsafe-safety-profile), in *Proceedings of the 26th Meeting of GI-FG TAV*, Stuttgart, Germany, 2007.
[33]. R. Pigan, M. Metter, Automatisieren mit Profinet (Automation with profinet), *Publics Corporate Publishing*, 2008.
[34]. F. Haslhofer, OpenSafety for wired and wireless connections over MQTT, MD Thesis, *Johannes Kepler University*, Linz, Austria, 2020.
[35]. B+R Automation, OpenSafety, Bernecker & Rainer, https://www.br-automation.com/enus/ technologies/ opensafety/
[36]. P. Sanz, I. Val, A. Urkidi, et al., Safety related systems design methodology for wireless time-varying channels, in *Proceedings of 17th IEEE International Conference on Factory Communication Systems (WFCS'21)*, 2021, pp. 123-130.

[37]. H. D. Wacker, J., Boercsoek, Binomial and monotonic behavior of the probability of undetected error and the 2-r-bound, *WSEAS Transactions on Communication*, Vol. 7, 2008, pp. 188-197.
[38]. H. D. Wacker, P. Pendli, J. Börcsök, Data transmission via erasure type channels protected by linear codes, *Journal of Physics: Conference Series*, Vol. 364, 2012, 012058.
[39]. Ph. Koopmanm, Best CRC Polynomials, https://users.ece.cmu.edu/~koopman/crc/

Chapter 2
Network-wide Vehicle Localization Algorithm Based on MEMS Sensor Data

Takayoshi Yokota

2.1. Summary

Global Navigation Satellite System (GNSS) has been widely utilized as a method to accurately determine the position of a vehicle, but at present, the accuracy may be degraded depending on the radio wave reception conditions from the satellite. There have also been growing concerns about cyber-attacks on GNSS. We have been developing a method for estimating the position of a traveling vehicle using MEMS sensor data acquired using acceleration, gyro, and geomagnetic sensors. This algorithm has been evaluated in several fixed courses and obtained good results with a position error of less than 1 m. In the current study, we present a solution to extend this vehicle localization algorithm with MEMS sensor data by combining it with a map-matching algorithm in order to associate reference sensor data with road links. This will significantly reduce the computational time by avoiding exhaustive cross-correlation calculation to vast amount of sensor data.

2.2. Introduction

As part of our ongoing research, we have developed a vehicle localization algorithm based on micro-electromechanical system (MEMS) sensors and demonstrated the error of 0.37 m in a smooth traffic region and about 1.6 m in fluctuating traffic through evaluation tests on a flat road along a river in Tottori, Japan [3, 4]. However, these results were for evaluations on the same fixed road sections. In order to generalize our vehicle localization algorithm, it must be capable of handling arbitrary road network reference data. This chapter describes how to extends the algorithm by including road link information on

Takayoshi Yokota
Faculty of Information Design, Tokyo Information Design Professional University, 2-7-1 Komatsugawa, Edogawa-ku, Tokyo, 132-0034, Japan

which the reference vehicle is running within the reference sensor data in order to associate sensor data with road link data based on recently published papers [1, 2]. This extension of the algorithm will significantly reduce the cross-correlation calculation time by avoiding exhaustive comparisons with all sensor data. Instead, the cross-correlation calculations are limited to adjacent links only. By applying a map-matching algorithm, we can obtain not only the vehicle's location (latitude and longitude) from RTK-GNSS but also the specific road link on which the vehicle is traveling.

2.3. Basic Algorithm of Vehicle Localization

Terrain-based localization is an interesting research field that has emerged in recent years [1-16]. Most of the prior studies have focused on using the pitch rate of running vehicles as a feature of the road surface. The algorithm of the vehicle localization based on cross-correlation of MEMS senor data we developed is shown in Fig. 2.1. First, to determine the location of the evaluation vehicle, our algorithm estimates the relative time lag between the two vehicles at the same point on the target road. The cross-correlation function is used to decide whether the two vehicles are located at the same position: specifically, if we can find a time lag between the two vehicles that makes the cross-correlation function maximum, it is presumably the correct time lag. Once we obtain the time lag, the location of the evaluation vehicle is substituted by the location of the reference vehicle as based on the estimated time lag. We obtain a set of reference sensor data by MEMS sensors equipped on the reference vehicle and correlated it with the sensor data of the evaluation vehicle. The reference vehicle comes equipped with a high precision GNSS(RTK-GNSS) [17] and its precise location is recorded together with the MEMS sensor data sampled at 50 Hz. We operated the RTK-GNSS at 5 Hz and performed interpolation and over-sampling at 50 Hz to obtain the reference location data. Once the MEMS sensor data matches the reference sensor data, we can estimate the position of the evaluation vehicle by substituting the position of the reference vehicle at the time where the MEMS sensor data best matches. Assume the MEMS sensor data obtained by several vehicles, where the first time is denoted by $f(t)$ and the second time is $g(t)$. The cross-correlation function over a range of $k = 0 \ldots N-1$ is given by

$$R(i,j) = \sum_{k=0}^{N-1}\left\{\left(f_{i+j+k} - \bar{f}_{i+j}\right)\left(g_{i+k} - \bar{g}_i\right)\right\} / \sqrt{F \cdot G}, \tag{2.1}$$

where

$$F = \sum_{k=0}^{N-1}\left(f_{i+j+k} - \bar{f}_{i+j}\right)^2 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots, \tag{2.2}$$

$$G = \sum_{k=0}^{N-1}\left(g_{i+k} - \bar{g}_i\right)^2, \tag{2.3}$$

$$\bar{f}_{i+j} = \frac{1}{N}\sum_{k=0}^{N} f_{i+j+k}, \tag{2.4}$$

$$\bar{g}_i = \frac{1}{N}\sum_{k=0}^{N} g_{i+k} \tag{2.5}$$

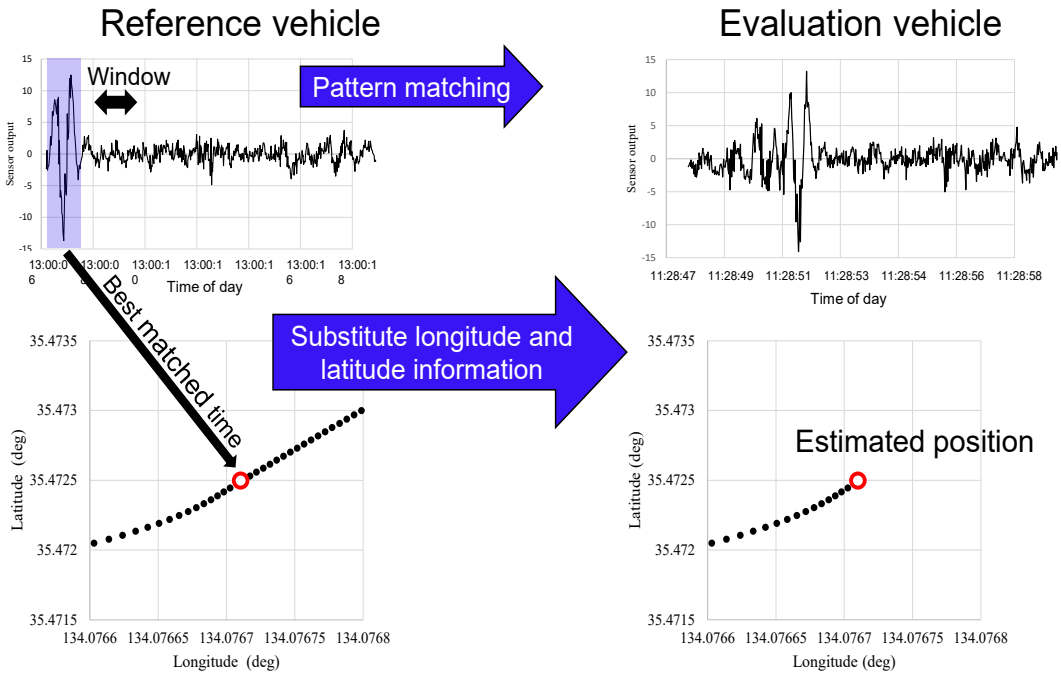


Fig. 2.1. Basic concept of vehicle localization algorithm based on MEMS sensor data.

If we can find the maximum value of the cross-correlation function, it means that the two vehicles exist at the same or very close location with the time lag. We used acceleration, gyroscope, and geomagnetic sensor (Mpu9250) [18] to obtain the sensor data. Each has three axes, so we have nine items of sensor data simultaneously. Each sensor data has a cross-correlation function, and we average all nine of them to obtain a stable cross-correlation function. For this purpose, we obtain the optimum weights by using a simulated annealing algorithm described in our previous work. [3, 4] The pitch rate has a maximum weight value 0.244. After the optimally weighted cross-correlation function is obtained we can estimate the time lag profile.

2.4. Target Area of the Evaluation

Fig. 2.2 shows the target area of about 3×7 km located in Tottori, Japan. It covers an almost completely flat area facing the Japan Sea and running alongside a river. Fig. 2.3 shows the corresponding satellite image. Fig. 2.4 shows the reference vehicle trajectory around node 42 together with its map-matched results (indicated by red dots).

This area was chosen because it is fairly flat and because it is difficult and challenging to localize a vehicle from the features of this road that are obtained by MEMS sensors.

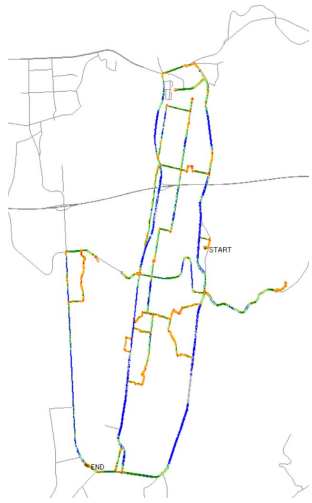


Fig. 2.2. Target area of about 3×7 km and reference vehicle trajectory. Reference sensor data of total milage of 31.8 km was acquired.

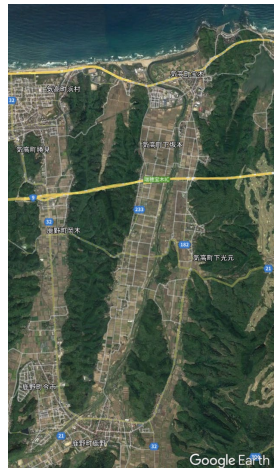


Fig. 2.3. Satellite image of 3×7 km target area in Tottori, Japan.

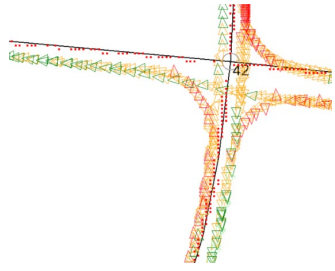


Fig. 2.4. Example of reference vehicle trajectories and its map-matched result. Reference vehicle trajectories around node 42 are shown. Map-matched results are indicated by red dots.

2.5. Vehicle Localization in a Series of Road Links

2.5.1. Map-matching Algorithm for Preprocessing the Reference Data

Map-matching algorithms are well-known in the field of vehicle navigation systems or vehicle route analysis in the logistic research field. Since the GNSS data of the reference vehicle is high-precision thanks to using RTK-GNSS, most of the anxiety about its accuracy is needless. We developed an in-house map-matching algorithm [10] that is highly precise even though the GNSS data interval is quite long (e.g., 10 seconds). For our application, we utilize the Japan Digital Roadmap as road links. For RTK-GNSS data that has a cm-order accuracy and a 5-Hz sampling interval of the reference vehicle, the error rate of the map-matching algorithm is usually 0 %.

Fig. 2.3 shows examples of the reference vehicle trajectories by RTK-GNSS and the results of map-matching, where red and green respectively indicate slow and moderate speeds. With this map-matching, we can determine which road link the reference vehicle was running on and at what time. This means that each time series of the MEMS sensor data of the reference vehicle can be associated with each road link. Colored rectangles indicate the positions of the reference vehicle along with the direction in which it is heading. Small red dots are map-matched points that are projected to appropriate positions on road links. The map-matching algorithm [10] has a grid resolution of one meter, so the red points are plotted on 1-m grid points.

2.5.2. Vehicle Localization Traversing a Road Link

The sensor data is a time series and the amount of it will continue to grow, so it needs to be organized in a sophisticated way. The sensor data are assigned to road link data by a map-matching algorithm. Fig. 2.5 shows the ground-truth trajectory of a reference vehicle by RTK-GNSS. Numbers indicated on the map are node numbers of the digital road map. In this case, each road link is denoted by a number (e.g., 42_10174), which is a non-directed link. The evaluation vehicle proceeded in a north-west direction via nodes 10175, 42, 10174, and 10164. When traversing node 42, the vehicle localization algorithm [1-4] must compare the sensor data time series of the evaluation vehicle with those of associated neighbor links. For example, sensor data associated with links 42-10174, 42-10165, and 42-10175 are candidates that must be evaluated. Thanks to the digital road map data, there is no need to perform an exhaustive comparison to all sensor data stored in the database.

2.5.3. Associating Reference Sensor Data with Road Links

Fig. 2.6 shows the data structure for associating reference sensor data with road links. The system always has a current link id, so by following the current link_id table, it can tell where associated sensor data exists, and which are next candidate links. By defining this data structure, the cross-correlation calculation-based vehicle localization algorithm can be applied to a general road network as long as the reference sensor data exists. In this

way, we can effectively focus on several candidate sensor data items at the same time – usually three per intersection, if we omit the rare case of a U-turn maneuver.

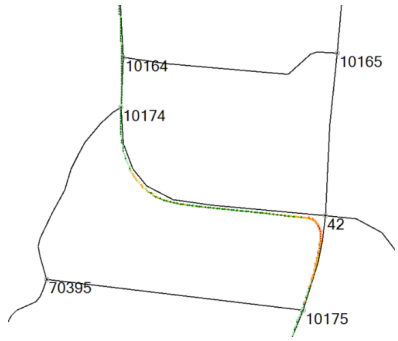


Fig. 2.5. The ground-truth trajectory of a reference vehicle by RTK-GNSS. Numbers indicated on the map are node numbers of the digital road map.

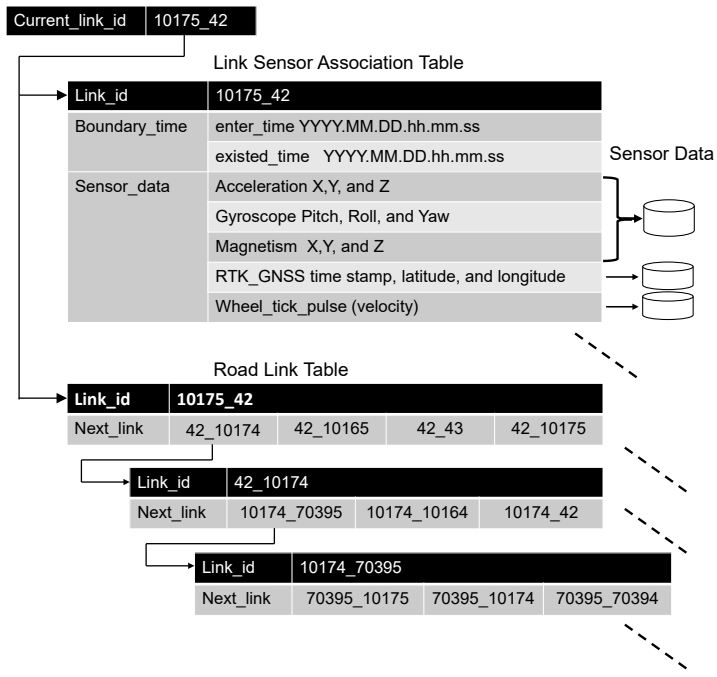


Fig. 2.6. Data structure for associating sensor data with road links.

Fig. 2.7 shows a screen shot of the Visual C++ 2022 software debugger, where six trips that passed through link 42_10175 are extracted: 2:13:24–2:13:33.6, 2:30:23.8–2:20:35.6, 2:45:41.6-2–45:48.2, 2:57:26.8–2:57:35.0, 3:20:18.4–3:20:19.2, and 3:47:48.8–3:47:55.4 (see the purple part). These are in Greenwich Mean Time (GMT), which is nine hours behind Japan Standard Time (JST). The travel time of each trip is 9.6 s, 11.6 s, 6.6 s, 8.2 s, 0.8 s, and 7.1 s.

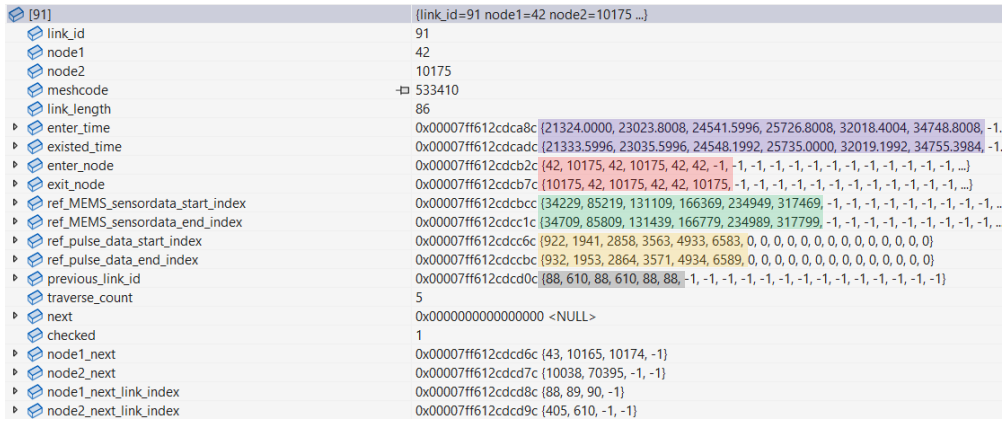


Fig. 2.7. Example of link management table. This is a screen shot of the Visual C++2020 debugger showing the trip information of the reference vehicle along link 42_1015. This is the implementation of the data structure shown in Fig. 2.5.

Among these, the travel time of 3:20:18.4–3:20:19.2, 0.8 s, is too short because of a miss-matching of the map-matching algorithm. Note that every trip except this trip took the same node 42 as the enter node and exit node (see the red part in Fig. 2.5). Therefore, this trip is ignored. Using the same procedure, we examined all links to determine how many trips existed on each link and ignored any that were too short. Other links of the target area are processed in the same manner. This process extracts the information of what time the reference vehicle entered and exited each link. The time resolution is that of the RTK-GNSS equipped on the reference vehicle and the time resolution is 0.2 s. This resolution could possibly be improved by interpolation, but since map-matching processing is time-consuming for higher resolution position data, it is kept to 0.2 s here. The total milage of the reference data is 31.8 km.

2.5.4. Assigning MEMS Sensor and Wheel Pulse Data to Each Trip Traversing a Link

The sensor data is a time series of nine axis data corresponding to acceleration, angular velocity, and geomagnetism for three axes of x, y, and z. As in the processing in the previous procedure, the beginning and the end of the record number of the MEMS sensor data are extracted by comparing the time stamp of the enter time and exit time of the link data in Fig. 2.7 (see the purple part) to those of the MEMS sensor data. For example, the second and fourth trips, which entered node 10175 and exited node 42 of the reference vehicle, started at 2:30:23.8 and ended at 2:30:35.6 and started at 2:57:26.8 and ended at 2:57:35.0 These trips are shown in Fig. 2.8(a) and (b).

These trips correspond to the MEMS sensor data that starts at index 85219 and ends at 85809 and starts at 166369 and ends at 166779 (see the green part of Fig. 2.7). The profiles of MEMS sensor data are shown Fig. 2.9(a) and (b), which correspond to the trips in Fig. 2.8(a) and (b). “Time of day” is depicted in seconds, as calculated by

$hour \times 3600 + minute \times 60 + sec$. From these results, the sensor data shows rather different aspects depending on whether the vehicle is running straight or turning when the link is an intersection link.

(a) 2:30:23.8–2:20:35.6 (b) 2:57:26.8–2:57:35.0

Fig. 2.8. Examples of reference vehicle trips over a link.

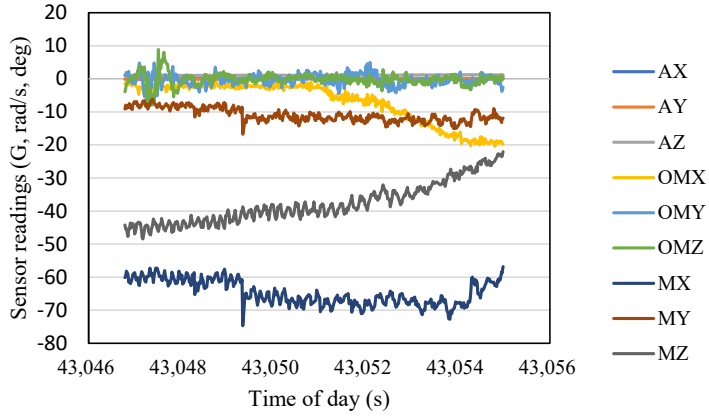


Fig. 2.9. (a) MEMS sensor data of reference vehicle 2:30:23.8–2:20:35.6.

Another important sensor is the wheel tick pulse, which serves as a velocity sensor when GNSS is not available. It might be possible to integrate the acceleration sensor to estimate the velocity, but such integration is difficult because eliminating the effect of gravity is extremely difficult. The corresponding wheel pulse sensor data indices are extracted from

1941 to 1953 and from 3563 to 3571 (see yellow part in Fig. 2.7). The profiles of the velocity obtained by the wheel pulse data are shown in Fig. 2.10(a) and (b).

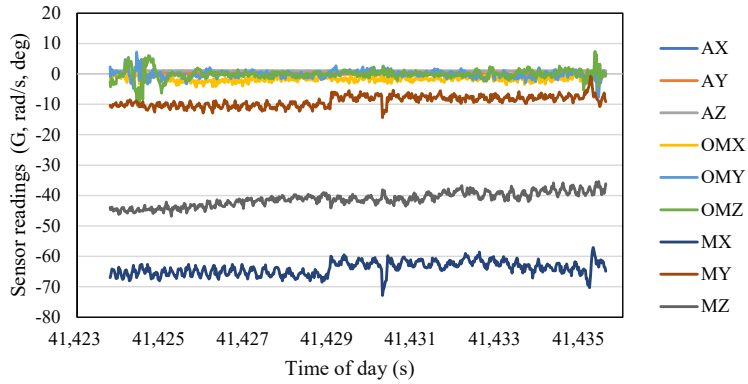


Fig. 2.9. (b) MEMS sensor data of reference vehicle 2:57:26.8–2:57:35.0.

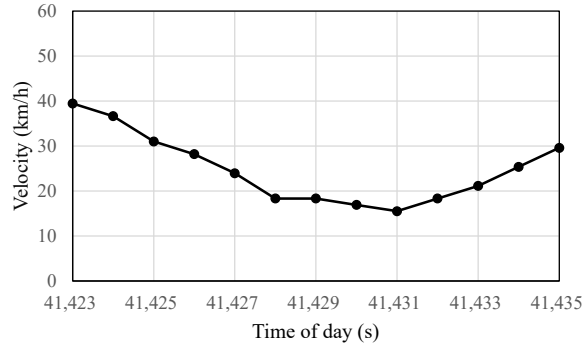


Fig. 2.10. (a) Velocity by wheel pulse data of the reference vehicle (index 1941 to 1953).

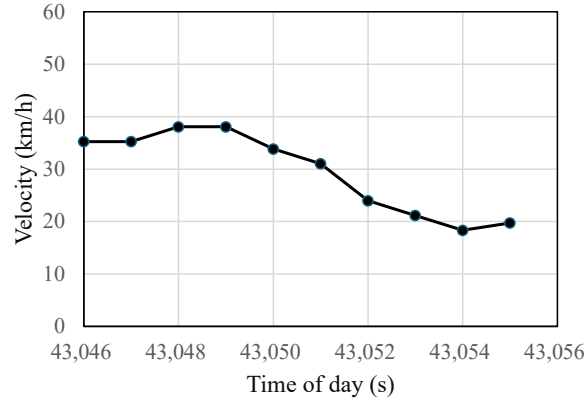


Fig. 2.10. (b) Velocity by wheel pulse data of the reference vehicle (index 3563 to 3571).

2.6. Processing of Evaluation Vehicle Data and its Results

2.6.1. Velocity Compensation

We obtained almost the same amount of evaluation vehicle data as reference vehicle data, but in this paper, only a portion of the evaluation data is evaluated for simplicity (the overall evaluation will be reported in the near future). The trajectory of a sample evaluation vehicle data is shown in Fig. 2.11. The corresponding MEMS sensor data of this trajectory is shown in Fig. 2.12 and its velocity profile in Fig. 2.14. As shown in Fig. 2.11, the evaluation vehicle ran through six links via nodes 10038, 10175, 42, 10174, 10164, 10168, and 10064. The travel time was 78 s, starting at 4:31:35.0 and reaching node 10064 at 4:32:53.0. This evaluation vehicle sensor data has the most significant cross-correlation with the reference data that ran through the same path (through nodes 10038, 10175, 42, 10174, 10164, 10168, and 10064), starting from 11:57:12.0 and ending at 11:58.26.0, whose series of MEMS sensor data is shown in Fig. 2.14 and its velocity profile in Fig. 2.15.

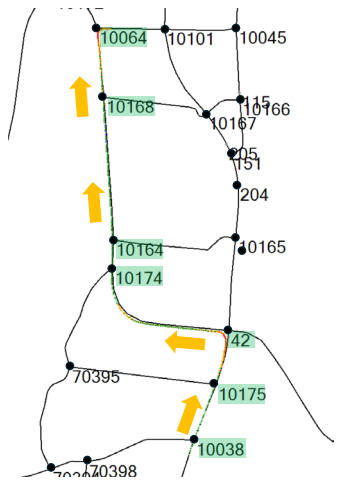


Fig. 2.11. Example of evaluation vehicle trajectory.

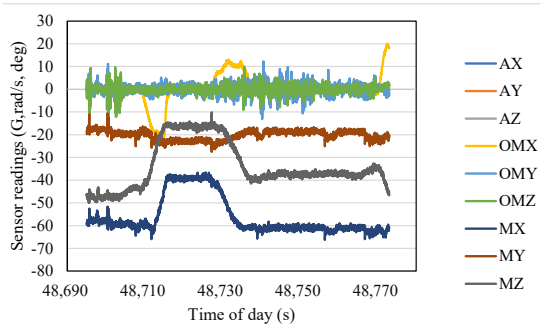


Fig. 2.12. MEMS sensor data of evaluation vehicle.

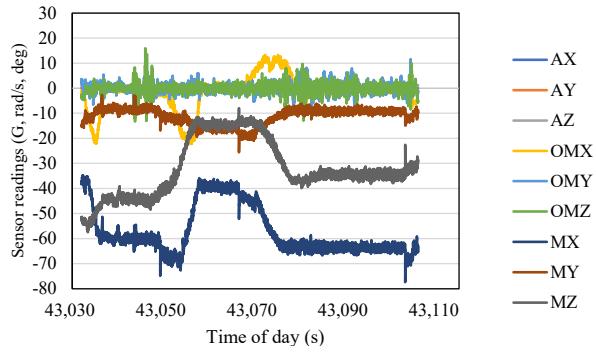


Fig. 2.13. Best matched MEMS sensor data of reference vehicle.

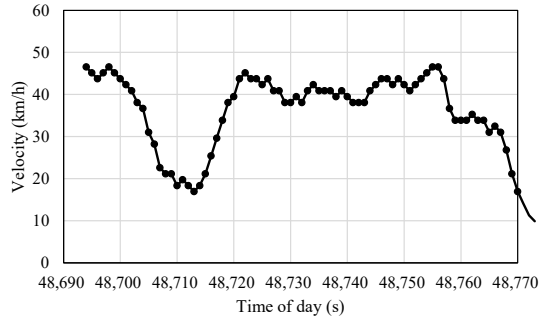


Fig. 2.14. Velocity profile of evaluation vehicle from Fig. 2.9.

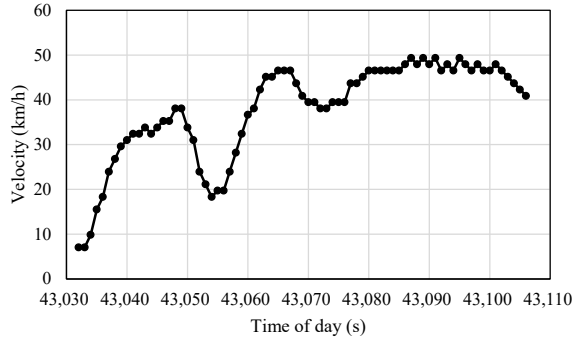


Fig. 2.15. Velocity profile corresponding to reference MEMS sensor in Fig. 2.11.

Among a number of combinations of link-separated sensor data, for the evaluation vehicle run shown in Fig. 2.11, sensor data along the series of links l0175_42, 42_10174, and 10174_10164 (same as that of the evaluation vehicle) gave the cross-correlation better than those obtained by other candidate sensor data. Comparing Fig. 2.14 and Fig. 2.15, no negligible velocity difference was observed, which makes the correlation calculation erroneous because the cross-correlation calculation assumes constant velocity during trips.

Because the velocity profiles of the reference vehicle and that of the evaluation vehicles is quite different, it is necessary to compensate the effect of velocity variation in the correlation calculation. Here, the compensation of the velocity variation is described.

Fig. 2.16 shows the velocity profile of the evaluation vehicle for 5 minutes including the time range of Fig. 2.14. We apply a velocity compensation algorithm where the virtual time axis $\varphi(t)$ is defined as

$$\varphi(t) = \int_{\tau=0}^{t} \frac{v(\tau)}{v_s} d\tau \tag{2.6}$$

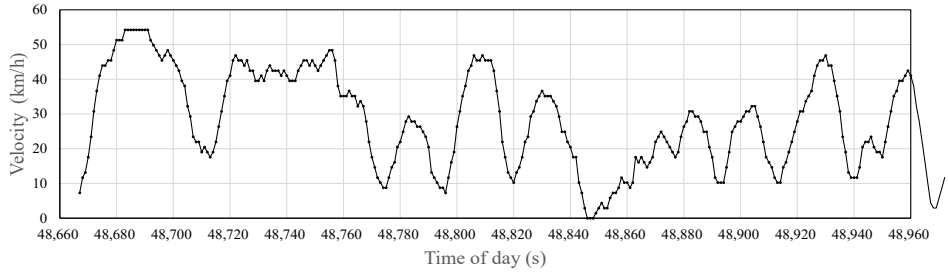


Fig. 2.16. Velocity profile corresponding to evaluation vehicle in Fig. 2.11.

The vehicle velocity (τ) can be easily obtained by multiplying wheel tick pulse by a scale factor. Of course, *φ(t)* becomes linear to *t* if the velocity profile $v(\tau)$ is a constant value. $\varphi(t)$is the time axis assuming the vehicle runs at the standard velocity v_s.

The virtual time axis *φ(t)* transforms the time series of the sensor data as if the vehicle was traveling at the standard velocity v_s. Here, v_s is set to 50 km/h. The high number of travel time is related to the time of day which is calculated as

$$\text{Time of day} = 3600 * \textit{hour} + 60 * \textit{minute} + \textit{sec}. \tag{2.7}$$

This represents a day as 86400 s. In estimating time lags, high number of seconds was properly subtracted as a bias.

For reference data obtained over a long period of time, Unix time may need to be used. If the velocity compensation algorithm is applied to the MEMS sensor data, one example of the evaluation vehicle sensor data (OMZ: pitch rate) shown in Fig. 2.17 is transformed into that shown in Fig. 2.18. The time axis is compressed or extended according to Eq. (2.6).

2.6.2. Resampling

The next step is resampling. After velocity compensation, the sensor data is no longer an equally sampled time series. Therefore, the sensor data needs to be resampled. We performed resampling using the nearest neighbor criteria, and the results were quite

acceptable. The comparison between the compensated and resampled sensor data is shown in Fig. 2.19. We consider the slight shift in the data not to be a serious issue in correlation calculations.

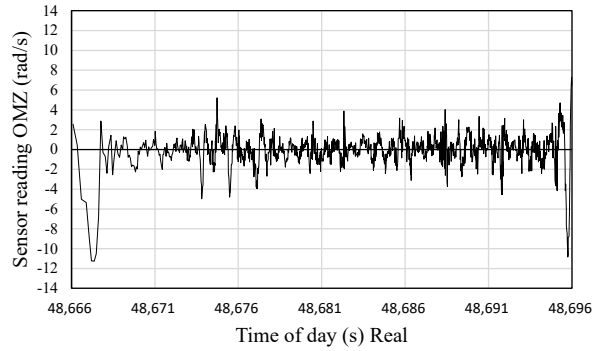


Fig. 2.17. The sensor data (OMZ) corresponding to the evaluation vehicle in Fig. 2.11 after applying velocity compensation.

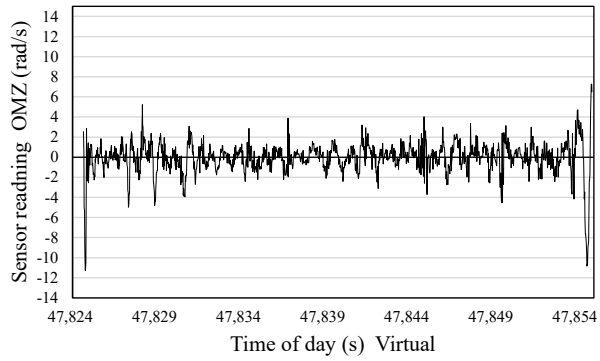


Fig. 2.18. The sensor data (OMZ) corresponding to the evaluation vehicle in Fig. 2.11 after applying velocity compensation.

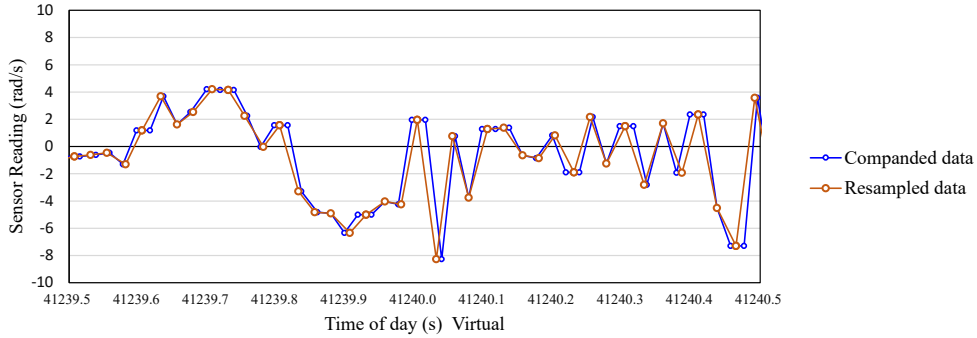


Fig. 2.19. The comparison between the compensated and resampled sensor data.

2.6.3. Time Difference Estimation and Vehicle Localization Results

The time difference chart (cross-correlation function) derived from calculating the correlation function for the velocity-compensated MEMS sensor data is shown in Fig. 2.20. A bright line clearly appears at around 20 seconds throughout the evaluation vehicle's travel time on the virtual time axis, despite some noise. If the velocity compensation and resampling were error-free, the bright line in Fig. 2.20 would have been a horizontal line.

Fig. 2.21 shows the plots of maximum cross-correlation at each travel time and Fig. 2.22 shows the estimated time lag after applying the noise reduction filter.

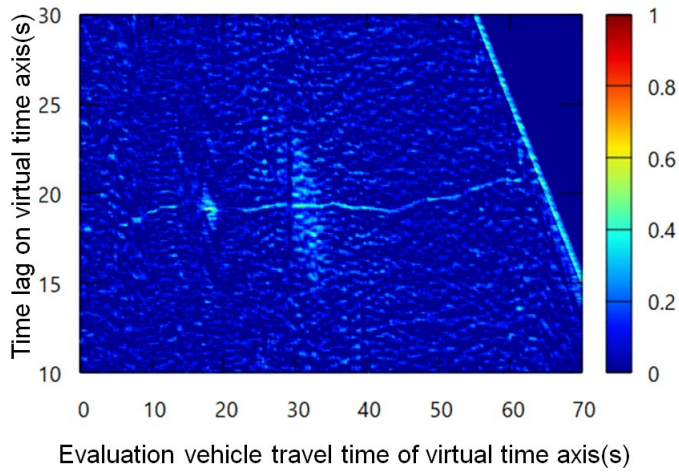


Fig. 2.20. The time difference chart (cross-correlation function) with velocity compensation.

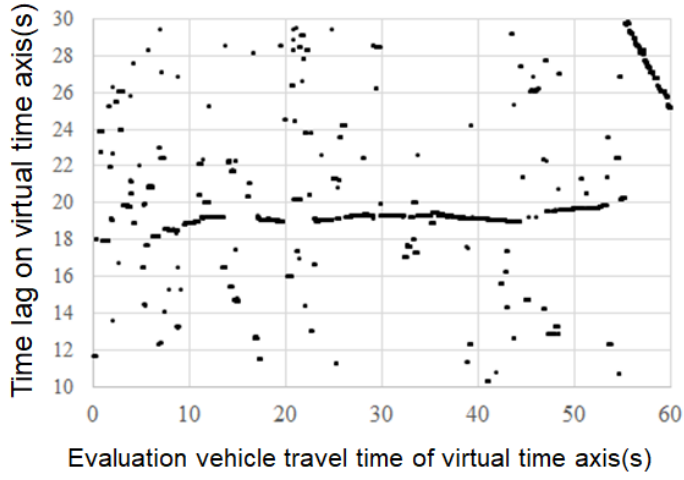


Fig. 2.21. Plots of maximum correlation at each travel time.

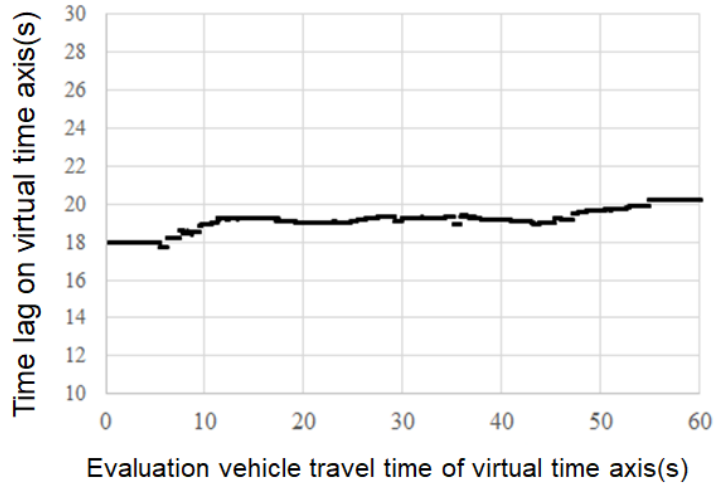


Fig. 2.22. Plots of maximum correlation after applying a noise reduction filter.

Fig. 2.23 shows the position error profile along the travel time of the evaluation vehicle. The mean squire error is 2.63 $m^2$and the root mean square error is 1.62 m. The most significant error is 6.5 m.

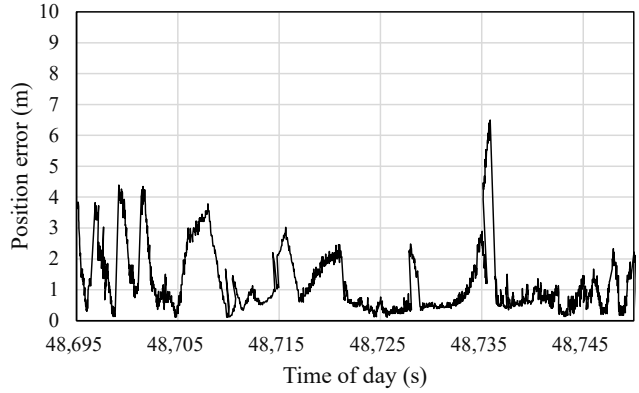


Fig. 2.23. Position error.

Fig. 2.24 shows the directional position error distribution. As we can see, the error distribution is dependent on the test course, and error in the direction of travel is dominant.

2.7. Conclusion

We have presented a solution to extend our vehicle localization algorithm using MEMS sensor data by combining it with a map-matching algorithm to associate reference sensor data to road links.

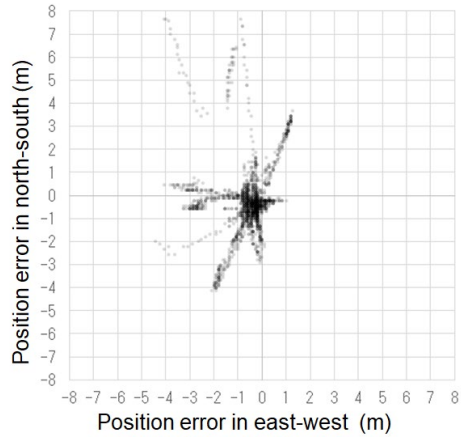


Fig. 2.24. Directional position error.

The following summarizes this research and our findings:

1) We devised a data structure that makes it possible to apply an algorithm that extracts road characteristics from acceleration, angular velocity, geomagnetism, and other sensors and estimates the position of a vehicle on a general road network. To this end, we developed a method that uses a map matching algorithm and digital road map information;
2) We found that MEMS sensor data, which is managed separately for each road link, needs to be kept for several cases, as the sensor data show different characteristics when entering or leaving that link by turning left or right;
3) Although it is necessary to connect several MEMS sensor data in a time series on time axis that is managed separately for each road link for candidate paths, the number of combinations is not considered too high. There are nine possible ways to enter and exit a link (each end by left turn, right turn, and straight run);
4) In this work, we only evaluated a sample evaluation vehicle run. The overall evaluation results will be reported in the near future;
5) The vehicle localization error would have been smaller if more accurate vehicle velocity had been obtained. For simplicity, we calculated the vehicle speed every second from the vehicle wheel tick pulses this time. However, if the vehicle speed were calculated continuously from the speed pulses, the time resolution of the speed would improve.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Number JP20K04723 and by the Japan Digital Road Map Association.

References

[1]. T. Yokota, Network-wide localization algorithm based on mems sensor data and its evaluation, *Sensors & Transducers*, Vol. 265, Issue 12, May 2024, pp. 17-26.

[2]. T. Yokota, Vehicle localization by correlated MEMS sensor data with velocity compensation, in *Proceedings of the IEEE 26th International Conference on Intelligent Transportation Systems (ITSC'23),* September 2023, pp. 3692-3698.

[3]. T. Yokota, Vehicle localization by optimally weighted use of cross-correlated MEMS sensor data, *Sensors & Transducers*, Vol. 261, Issue 2, 2023, pp. 1-9.

[4]. T. Yokota, T. Yamagiwa, Vehicle localization by optimally weighted use of MEMS sensor data, in *Proceedings of the 3rd IFSA Winter Conference on Automation, Robotics and Communications for Industry 4.0/5.0 (ARCI'23)*, Feb. 2023, pp. 79-84.

[5]. T. Yamagiwa, T. Yokota, Vehicle localization method based on MEMS sensor data comprising pressure, acceleration and angular velocity, in *Proceedings of the IEEE International Conference on Mechatronics and Automation (ICMA'22)*, Aug. 2022, pp. 786-791.

[6]. T. Yokota, Vehicle localization by altitude data matching in spatial domain and its fusion with dead reckoning, *International Journal of Mechatronics and Automation*, Vol. 8, Issue 4, Jan. 2021, pp. 208-216.

[7]. T. Yokota, Vehicle localization by dynamic programming from altitude and yaw rate time series acquired by MEMS sensor, *SICE Journal of Control, Measurement, and System Integration*, Vol. 14. Issue 1, April 2021, pp. 78-88.

[8]. T. Yokota, Vehicle localization based on MEMS sensor data, in *Proceedings of the 60th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE'21)*, Sept. 2021, pp. 1094-1100.

[9]. T. Yokota, Localization algorithm based on altitude time series in GNSS-denied environment, in *Proceedings of the 59th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE'20)*, September 23-26, 2020, pp. 952-957.

[10]. T. Yokota, M. Okude, T. Sakamoto, R. Kitahara, Fast and robust map-matching algorithm based on a global measure and dynamic programming for sparse probe data, *IET Intelligent Transport Systems*, Vol. 13, Issue 11, 2019, pp. 1613-1623.

[11]. J. Tsurushiro, T. Nagaosa, Vehicle localization using its vibration caused by road surface roughness, in *Proceedings of the IEEE International Conference on Vehicular Electronics and Safety*, Yokohama, Japan, Nov. 2015, pp. 164-169.

[12]. A. J. Dean, R. D. Martini, S. N. Brennan, Terrain-based Road vehicle localization using particle filters, *International Journal of Vehicle Mechanics and Mobility*, Vol. 49, Issue 8, 2011, pp. 1209-1223.

[13]. E. Laftchiev, C. Lagoa, S. Brennan, Terrain-based vehicle localization from real-time data using dynamical models, in *Proceedings of the 51st IEEE Conference on Decision and Control (CDC'12)*, Maui, HI, USA, 2012, pp. 3366-3371.

[14]. J. Gim, C. Ahn, Ground feature-based vehicle positioning, in *Proceedings of the Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE'20)*, 2020, pp. 983-984.

[15]. J. Gim, C. Ahn, IMU-based virtual road profile sensor for vehicle localization, *Sensors*, Vol. 18, Issue 10, 2018, pp. 33-44.

[16]. X. Qu, B. Soheilian, N. Paparoditis, Landmark based localization in urban environment, *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 140, June 2019, pp. 90-103.

[17]. ZED-F9P module, u-blox F9 high precision GNSS module, https://www.u-blox.com/en/product/zed-f9p-module

[18]. MPU-9250, Nine-Axis (Gyro + Accelerometer + Compass) MEMS MotionTracking™ Device, https://invensense.tdk.com/products/motion-tracking/9-axis/mpu-9250/

Chapter 3
Channel Shortening and Viterbi Algorithm Based Equalization for High Data Rate Baseband Communication Over a Frequency-selective Channel

Miqueu Paul, Fabrice Belvèze, Jean-Marc Brossier and Laurent Ros

3.1. Introduction

When the frequency response of a channel is very different from that of the perfect (distortion-free) channel, transverse filtering may only be able to equalize the channel very imperfectly in a practical implementation. The transverse filter may need to have a very large number of coefficients, and it may also amplify noise significantly, resulting in high Bit Error Rate (BER) despite the elimination of Inter-Symbol Interference (ISI). It is in this context that nonlinear equalization solutions prove very useful for better symbol detection.

In 1979, C.A. Belfiore presented a history of non-linear equalization methods in [1] and concluded that the Maximum Likelihood Sequence Estimator (MLSE) offers the best compromise between performance, complexity, and implementation difficulty. This equalizer, based in practice on the Viterbi Algorithm (VA), initially proposed for convolutional code decoding [2, 3], was proposed simultaneously by Forney [4], Omura [5] and Kobayashi [6]. In a much more recent publication focused on SerDes technology, the article [7] presents a very comprehensive state of the art of nonlinear equalization adapted to very high-speed serial links. Part of the state of the art combines Decision Feedback Equalization (DFE) and VA.

In the rest of this work, the VA will therefore be at the center of our receiver equalization strategy. This choice is corroborated by the state of the art, where more and more of the

Miqueu Paul
Gipsa-Lab, Université Grenoble Alpes, CNRS, Saint Martin d'Hères, 38400, France

latest communication systems integrate VA at receiver. The second section introduces this MLSE block, detailing its operation and main characteristics.

As the computational complexity of the algorithm is critical, we then focus from Sections 3.3 to 3.6 on channel shortening upstream of VA by studying, in this context, state-of-the-art linear pre-filtering techniques and non-linear equalization techniques.

Finally, we compare in Section 3.7, for each of the channel shortening methods presented, the performance of the communication system for a real-life frequency selective channel proposed by Richard Mellitz in [8] during an IEEE 802.3df working group meeting. This working group aims at establishing a standard for all communication systems using the Ethernet protocol for high data rates (above 100 Gbps).

This comparison part reports on our contributions in the field of nonlinear equalization of very high-speed serial links. An interpretation of the performance of each receiver is proposed, taking into account the advantages and disadvantages inherent in each equalization block. We conclude in Section 3.8, in light of the specifications given by the standard, on the equalization method best suited to very high-speed serial links.

This chapter is an extension of the paper entitled “Miqueu, P., Belveze, F., Brossier, J. M., & Ros, L. (2025, February). Non-linear equalization techniques for high data rates serial links.” presented at the ARCI’2025 international conference, Automation, Robotics & Communications for Industry 4.0/5.0, p. 154.

3.2. The Maximum Likelihood Sequence Estimator (MLSE) Using the Viterbi Algorithm

Before Forney proposed in [4], concurrently with Omura and Kobayashi in [5, 6] and [9], to reuse the VA [2, 3], initially designed for convolutional code decoding, in the context of equalization, the naivest method of sequence detection was to calculate the likelihood of all possible symbol sequences based on observations. As the complexity increased exponentially with the size of the symbol sequence considered, sequence detection in the sense of maximum likelihood was abandoned. MLSE, based on the VA, offers an optimal solution to the problem of estimating a sequence of symbols for a finite global channel impulse response with complexity independent of the size of the sequence of symbols to be estimated. It evolves according to the size of the global channel impulse response at the input of the MLSE block.

Readers interested in understanding the reasoning behind the development of this algorithm and the details of its computations can refer to Forney’s pioneering article [4], in which the VA is used for the first time to equalize a channel. Here, we only introduce the algorithm’s features relevant when considering high data rate serial link equalization.

3.2.1. The Viterbi Algorithm Applied to Equalization

The block scheme of the communication system we consider in this chapter is illustrated Fig. 3.1.

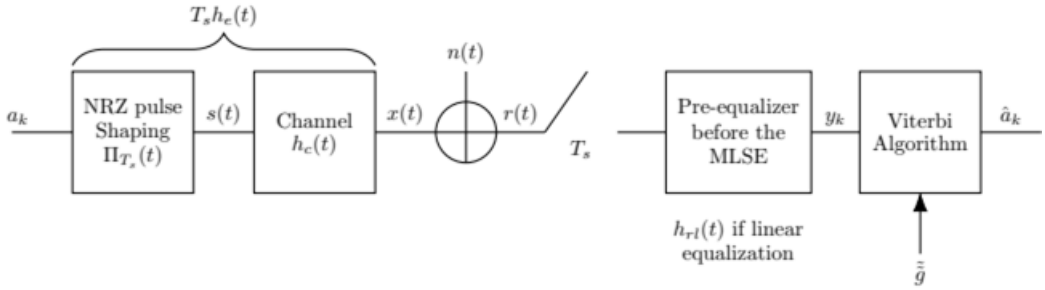


Fig. 3.1. Block scheme of the communication system with a MLSE at the receiver.

The analog received signal is $r(t) = T_s.\sum a_k h_e(t - kT_s)$ where a_k are the M-ary PAM symbols, h_e is the transmit impulse response, including rectangular symbol pulseshape $\Pi_{T_s}(t)$(unit amplitude during one symbol duration T_s) convolved by the channel impulse response $h_c(t)$ (i.e. $h_e = \Pi_{T_s} * h_c$) and $n(t)$ is AWG noise with single-sided power spectral density N_0.

The VA takes as a parameter the partially equalized impulse response of the global channel denoted by $\tilde{\tilde{g}}$. The impulse response of the global channel, in the case where all the blocks that make up the communication system in Fig. 3.1 are filters, can be expressed as a convolution product such as:

$$\tilde{\tilde{g}}(t) = [\tfrac{1}{T_s}\Pi_{T_s} * h_c * h_{rl}](t), \tag{3.1}$$

where $h_{rl}(t)$ is the impulse response of the linear reception filter before the MLSE. If the signal processing before the MLSE is not linear, the overall channel cannot be expressed as a convolution product. In particular, in the case where pre-equalization is provided by a DFE, we will construct a pseudo impulse response, the construction of which will be detailed later in Section 3.4.

To be complete on the expression of the impulse response of the global channel at the VA input, we will say that in the case where it is not of finite size, it is truncated while trying to conserve as much energy as possible. If the coefficients of the global channel impulse response are unknown or vary slowly during transmission (which is not the case considered here), algorithms can be used to obtain them and track small variations [10, 11].

The VA algorithm will use a discrete-time version of the overall channel:

$$\tilde{\tilde{g}}_k = T_s\tilde{\tilde{g}}(t)|_{t\,=\,(k_0+k)T_s}, \tag{3.2}$$

where k_0 is the delay (expressed in number of symbols) for sampling.

Therefore, considering a pseudo impulse response $\tilde{\tilde{g}}$ with $\nu + 1$ coefficients at the VA input and a zero-symbol propagation delay in the communication chain ($k_0 = 0$), the samples at the VA input can be expressed as:

$$y_k = \lambda a_k + ISI_k + b_k, \tag{3.3}$$

where $\lambda = \tilde{\tilde{g}}_0$ and the ISI term can be expressed as

$$ISI_k = \sum_{n=k-\nu}^{k} a_n \tilde{\tilde{g}}_{k-n} \tag{3.4}$$

Since the impulse response of the overall channel is finite, we call the finite sequence of symbols participating in the ISI term at time k the memory configuration of the channel.

The memory configuration changes, resulting from symbol transmission, form a sequence for which the Viterbi algorithm provides an estimate that maximizes the likelihood in presence of Additive White Gaussian Noise (AWGN) and for a frequency selective channel [12].

Now that the relevant VA features have been introduced, we rise the reader's attention on several key points that have to be considered for high data rate serial link equalization.

3.2.2. To be Remembered in the Following

Firstly, although less complex than an exhaustive search for the sequence that maximizes likelihood, VA remains computationally complex. This is its main drawback. If the channel impulse response has $\nu + 1$ coefficients and the transmission alphabet contains M different symbols, the complexity of the VA will be $M^{\nu+1}$. The complexity increases exponentially with the size of the channel impulse response, making the use of the VA alone unfeasible in many practical cases. The prospect of taking advantage of the optimal equalization algorithm has generated a wealth of literature on different ways to reduce its complexity.

Secondly, VA requires knowledge of the impulse response.

Thirdly, in order to make a reliable decision on a symbol, VA needs to process τ samples after the sample of interest. This is the number of transitions in the trellis on which we traceback the sequence of symbols that maximizes the likelihood. This is a parameter that allows the reliability of the estimated symbol to vary. The deeper the backtracking, the greater the probability that the paths have converged to a single common ancestor, which guarantees the optimality of the decision. The backtracking length τ depends on the size of the impulse response of the overall channel at the input of the algorithm, denoted $\nu + 1$. We take $\tau = 5\nu$.

In order to limit computational complexity, which is the main challenge when it comes to using the VA for equalization, we have chosen to focus on channel shortening.

3.3. Channel Shortening Using Linear Filtering

While we can obtain the expression of the transverse filter for an ISI elimination criterion, linear filtering upstream of the VA pre-processes the signal to optimize VA performance. In some of the literature, this pre-filter is a whitened matched filter that ensures the best Signal to Noise Ratio (SNR) and white thermal noise at its output, which minimizes the probability of error at the VA output [4]. We thus first introduce a communication system with a whitened matched filter and consider it as a reference. In other literature, it is shown that colored noise at the VA input has little effect on the error probability [19]. We will thus next consider another pre-filter that seeks to limit the algorithmic complexity of the VA by shortening the channel impulse response as reducing algorithmic complexity is critical when implementing the VA.

3.3.1. Whitened Matched Filter

Following Forney's pioneering article [4], many methods for shortening the channel impulse response have emerged, each with its own advantages and disadvantages. Most articles in the literature consider that the receiver is equipped with a whitened matched filter at its input, ensuring that the signal is a sufficient statistic for determining the symbols sent. It is demonstrated in [13] for many performance criteria such as BER, BER after ISI equalization, and SNR, that adding a whitened matched filter does not degrade performance. We thus consider this communication system as a reference and begin by considering a receiver equipped with a whitened matched filter at its input.

The block linear analog Rx filter followed by a sampler in Fig. 3.2 can then be broken down into an analog filter matched to the overall channel, a synchronous sampler at symbol rate, a digital whitening filter for thermal noise, and a digital channel shortening filter, as illustrated in Fig. 3.3.

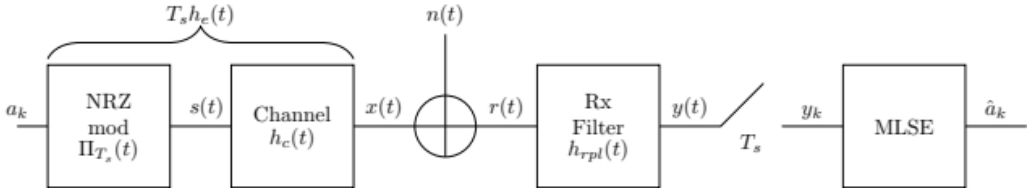


Fig. 3.2. Communication system with a linear filter with impulse response h_{rpl} at Rx.

Computation of the whitened matched filter is well known in the literature and we refer the reader to it for further details [4].

We only raise the reader's attention to the fact that, using the good computation criteria, the whitened matched filter can minimize the channel impulse response phase [4]. The impulse response of a minimum phase filter has remarkable properties, as it is the impulse response with maximum energy distribution over the first samples (which is very important, as we will see in section 4) and it is causally invertible.

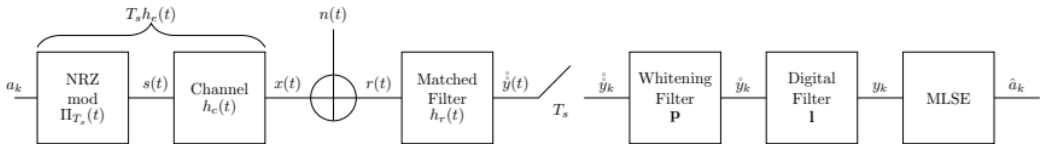


Fig. 3.3. Communication system with linear filters corresponding to the linear filter with impulse response h_{rpl} at Rx illustrated in Fig. 3.2.

We compare this reference communication system to another one, with a channel-shortening filter at the input of the MLSE, as it is supposed to reduce the computational complexity of the system at the price of a colored noise at the input of the MLSE.

3.3.2. Channel Shortening Using the Falconer and Magee's Filter

The purpose of shortening the impulse response of the channel upstream of the VA is to transform it into a Target Impulse Response (TIR) that is shorter than the initial impulse response in order to reduce the complexity of the VA.

3.3.2.1. Foreword on Channel Shortening

Various studies exist on channel shortening. Qureshi and Hall proposed in [11] an adaptive receiver, based on the work of Lucky [15], that aimed at shortening the channel impulse response before the Viterbi algorithm and minimizing thermal noise power. The authors recommend obtaining the TIR by truncating the channel impulse response, which limits the Mean Square Error (MSE) at the equalizer output [16]. An analytical study of MLSE performance is conducted based on the size and coefficients of the TIR using the flow-graph technique proposed by Forney in [4]. Considering performance comparisons carried out in the literature [19], we decided to adopt a method proposed by Falconer and Magee in [17], which we introduce here.

3.3.2.2. Falconer and Magee Filter Principle

Falconer and Magee propose in [17] a transverse filter that takes as arguments the sampled impulse response of the overall channel and the number of TIR coefficients. This solution has the advantage of providing the impulse response of the channel-shortening filter at the same time as the TIR that will be used in the Viterbi algorithm. Under the constraint that the TIR energy is normalized, Falconer and Magee's algorithm, illustrated in Fig. 3.4, minimizes the MSE between the output of the partial equalization filter for a given symbol sequence and the output of the TIR when fed with the given symbol sequence, while minimizing the thermal noise power. We refer the readers to [17] for computation details. The communication system with the channel shortening filter, noted as Falconer filter, is illustrated in Fig. 3.5. The whitened matched filter is described as optional because we implement the receiver with and without this filter without modifying the algorithm for

obtaining the Falconer filter coefficients. If the receiver does not have a matched filter at the input, the filter $h_{rp}(t)$ becomes the Kronecker delta function.

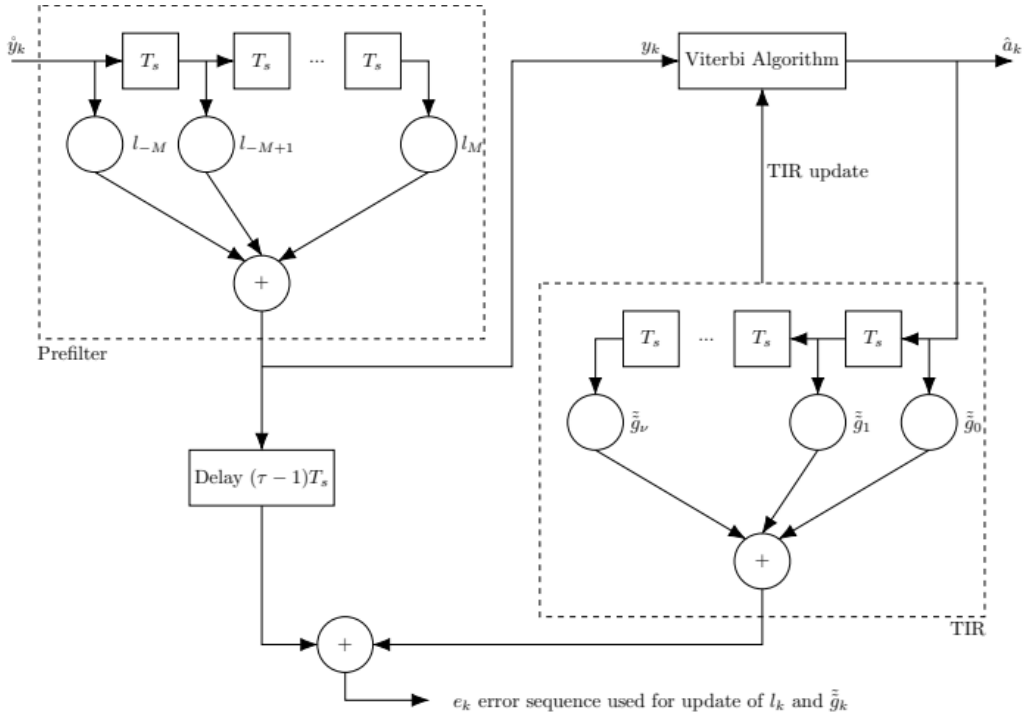


Fig. 3.4. Receiver's structure proposed by Falconer and Magee.

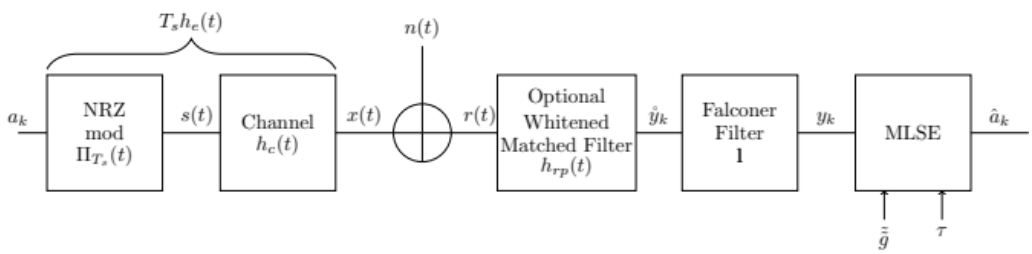


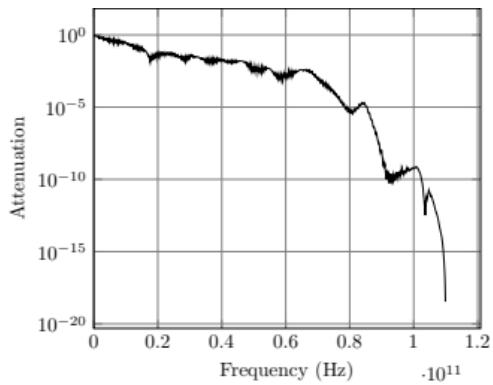
Fig. 3.5. Communication system with a *digital* transverse filter proposed by Falconer and Magee in [17] which taps are arranged in the vector l. Decision is taken by a MLSE. Its parameters are the TIR $\tilde{\tilde{g}}$ and the traceback depth τ of the VA.

Beare, in [19], notes that the VA is only optimal when the thermal noise at its input is white, a condition that is not met when using the method described in [11] or [17]. The author therefore proposes using the TIR whose spectrum is as close as possible to the spectrum of the original channel, given the limited size of the TIR. The author does not seek to minimize the power of the thermal noise at the output of the linear filter, and the

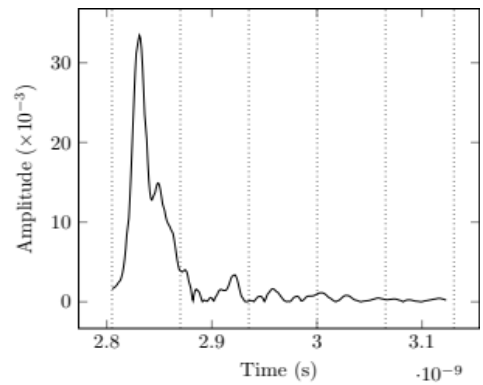
experimental results he obtains do not show any improvement in performance for the scenario of interest. We will therefore stick with Falconer's filter.

As part of our research, we implemented the Falconer and Magee filter, with and without a whitened matched filter, for the channel proposed by Richard Mellitz during a session of the IEEE 802.3df work group on the standard. Characteristics of various channels are available in [8]. We chose the 0.5-meter-long backplane link with 1 mm large SMA connectors and a PCB loss of 3.2 dB which S-parameters can be found in the sma_1.0mm_3.2dB_500mm_NVAC_thru.s4p file.

S-parameters for the considered as well as for other channels (various lengths, connectors or PCB loss) are freely available at [39]. We compute, from the S-parameters, the channel's impulse response using the FFT and an over-sampling ratio of 2^{10} and illustrate it in Fig. 3.6(b). The spectrum directly given by the S-parameters is illustrated in Fig. 3.6(a).



(a) Power spectral density of the channel proposed by Richard Mellitz in [8].



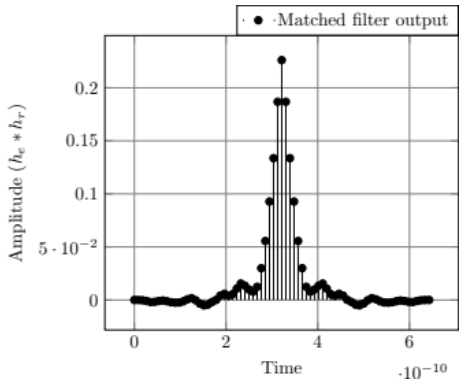
(b) Channel's I.R. The space in-between two dotted lines corresponds to a 10 symbols period gap for a 4-PAM and a binary data rate of 224 Gbps.

Fig. 3.6. Power spectral density and impulse response (I.R.) of the channel proposed by Richard Mellitz. It corresponds to a backplane link of length 500 mm linked to a chip by SMA connectors of 1.0 mm diameter and a 3.2 dB attenuation on the chip.

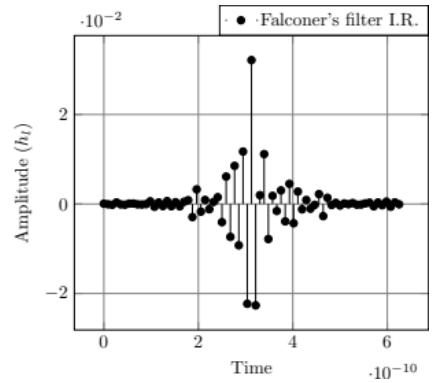
Are illustrated Fig. 3.7 for a 4-PAM, a TIR size of $\nu + 1 = 4$ coefficients, a Falconer filter of $N_{FM} = 2M + 1 = 71$ coefficients, and a receiver with a whitening filter at the input:

- The impulse response of the channel at the output of the matched filter alone $h_e * h_r(t)|_{t = kT_s}$;
- The impulse response of the Falconer filter $h_l(t))_{t = kT_s}$ for $N_{FM} = 71$ and $\nu + 1 = 4$ coefficients;

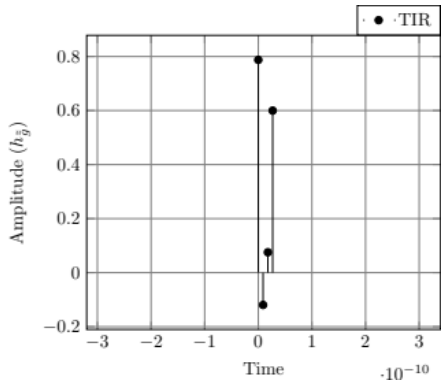
- The TIR $\tilde{\tilde{g}}(t))_{t = kT_s}$ for $N_{FM} = 71$ and $\nu + 1 = 4$ coefficients;
- The impulse response of the channel at the output of the Falconer filter $(h_{erp} * h_l)(t)|_{t = kT_s}$.



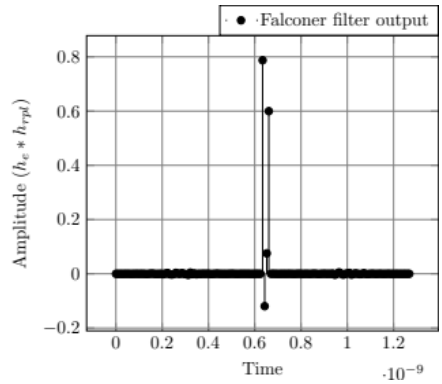
(a) Global channel I.R. at the matched filter output for a 4-PAM.



(b) Falconer's filter I.R. for $\nu + 1 = 4$, $N_{FM} = 71$ coefficients.



(c) TIR at the output of the Falconer's filter $\nu + 1 = 4$, $N_{FM} = 71$.

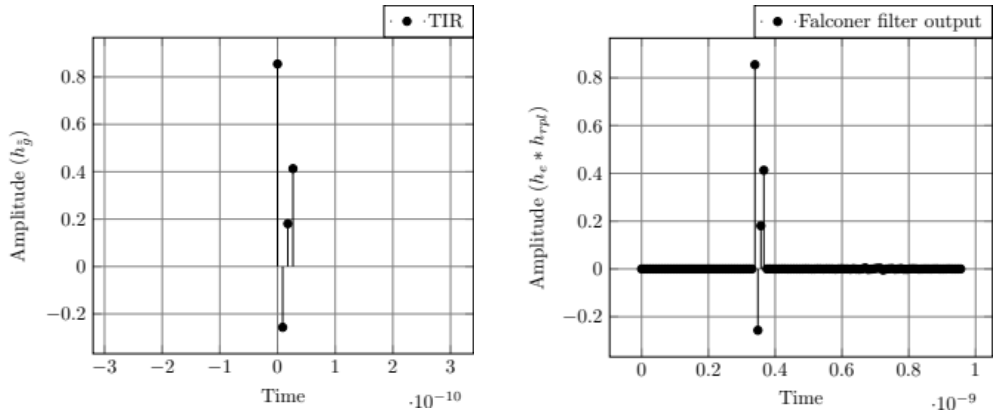


(d) Convolution between the global channel I.R. and the Falconer filter for $\nu + 1 = 4$, $N_{FM} = 71$ coefficients.

Fig. 3.7. Global channel at the output of the matched filter I.R. (a), Falconer's filter I.R. (a), TIR (c) and I.R. of the global channel at the output of the Falconer filter (d) for $\nu + 1 = 4$, $N_{FM} = 71$ with a matched filter at Rx.

We note that the impulse response of the overall channel shown in Fig. 3.7(d), considering the coefficients containing most of the signal energy and neglecting a little ISI, corresponds to the TIR in Fig. 3.7(c). This filter is therefore a good candidate for channel shortening, reducing the overall channel IR from approximately 26 to approximately 4 non-zero coefficients for 4-PAM.

Similarly, when there is no matched filter at the receiver, Figs. 3.8(a) and 3.8(b) illustrate the TIR and the impulse response of the overall channel with $\nu + 1 = 4$, $N_{FM} = 71$.



(a) TIR with $\nu + 1 = 4$, $N_{FM} = 71$.

(b) Overall channel with $\nu + 1 = 4$, $N_{FM} = 71$.

Fig. 3.8. TIR (a) and I.R. of the overall channel at the output of the Falconer's filter for $\nu + 1 = 4$, $N_{FM} = 71$ and without a matched filter at Rx.

As already mentioned earlier in this paper, thermal noise amplification is the main drawback of linear filtering. For performance comparison purpose, we thus turn to nonlinear equalization applied to channel shortening.

3.4. Channel Shortening Using Decision Feedback Equalizer (DFE)

The article by C.A. Belfiore [1] provides a very comprehensive overview of DFE. Initially presented as a suboptimal alternative to MLSE, DFE does indeed perform less well than MLSE, but its complexity varies linearly with the size of the overall channel impulse response. It also amplifies thermal noise less than a linear filter [20] but can be subject to error propagation. We first introduce the DFE as proposed in [21] and [22], which we refer to as the traditional DFE. Next, we explain how it can shorten, in the form of a partial DFE, the impulse response of the overall channel upstream of the MLSE, as proposed in [23].

3.4.1. Structure and Principle of a Traditional DFE

The classic structure of the receiver with DFE is illustrated in Fig. 3.9.

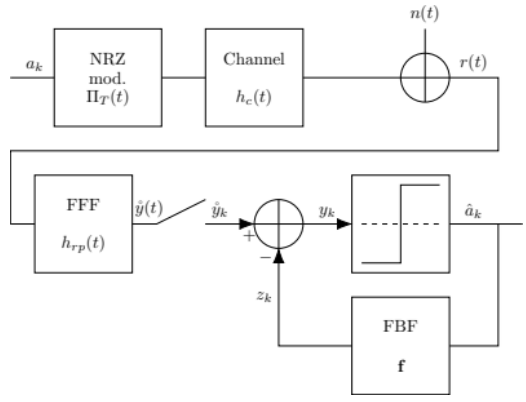


Fig. 3.9. Traditional structure of a DFE at Rx with Feed-Forward Filter (FFF) and FeedBack Filter (FBF).

The signal at the output of the channel is first filtered by the Feed-Forward Filter (FFF)

$$\mathring{y}(t) = [r * h_{rp}](t)$$

The impulse response of the output of the FFF is denoted by $\tilde{g}$ (that means $\tilde{g} = h_e * h_{rp}$). At the output of the FFF, the signal is sampled, then a sum of decisions weighted by the coefficients of the FBF filter stored in the vector denoted by f is subtracted from it, as shown in Fig. 3.9. The FBF filter hasN_{fbf}coefficients. We have:

$$f_n = \tilde{g}_{n+1} \text{for} n \in \{0, N_{fbf} - 1\}$$

We then have:

$$z_k = \sum_{n=0}^{N_{fbf-1}} f_n \hat{a}_{k-n+1} \tag{3.5}$$

where $\hat{a}_k$ are the estimates, at the output of the decision block, of the symbols sent. The sum expressed in equation (4), known as the post-cursor ISI, aims to cancel out the ISI generated by the symbols preceding the symbol of interest by subtraction at the output of the FFF. We then have:

$$y_k = \mathring{y}_k - z_k,$$

the signal samples on which the decision is made. The decision on the sample of interest y_k is then made by choosing the amplitude in the modulation dictionary that minimizes the MSE between the amplitude of z_k and the amplitudes in the modulation dictionary. The impulse response of the equivalent overall channel after subtraction of the post-cursor ISI is such that:

$$\tilde{\tilde{g}}_k = \begin{cases} \tilde{g}_0 \; for k = 0 \\ 0 \; otherwise \end{cases}$$

3.4.2. Calculation of FFF and FBF Coefficients

In the following and throughout the literature, the decisions made are assumed correct when they are used to determine the coefficients of the optimal FFF and FBF filters, for example. In other words: $\hat{a}_k = a_k$. The coefficients of the FFF and FBF can be calculated adaptively [24]. Since the channel is assumed known here, we calculate the coefficients analytically. There are three criteria for determining the coefficients of the FFF and FBF:

- Zero Forcing criterion at the DFE output [25];
- Minimum Mean Squared Error criterion between the decision made and the sample before the decision [26];
- Minimum Error Probability criterion at the DFE output [27].

It is demonstrated in [20] that the ZF criterion and the MMSE criterion are equivalent for the considered scheme. Also, as we only plan to use the DFE to shorten the input channel of a VA the third criterion is not relevant in our case. We therefore calculate the DFE coefficients using the ZF criterion, which, according to [28], makes the FFF the whitened matched filter that minimizes the channel phase.

3.4.2.1. DFE with FFF Whitened Matched Filter

Referring to the literature, we first implement the receiver with a whitened matched filter. The coefficients of the FFF and FBF, as well as the impulse response of the channel at different points of the DFE, are illustrated in Fig. 3.10.

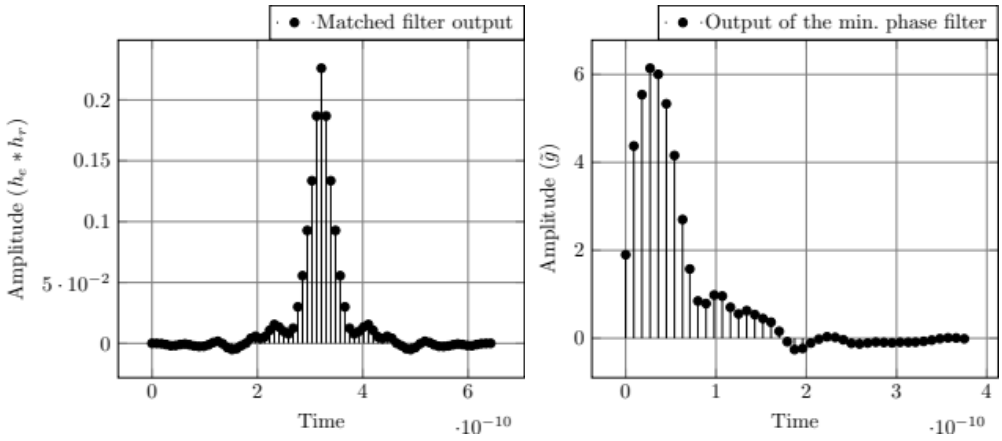
The FFF, which impulse response is denoted h_{rp}, is decomposed into an analog filter adapted to the overall transmission channel, which impulse response is denoted h_r, and a digital whitened filter that corrects the phase of the impulse response h_p. The impulse response $\tilde{g}$ of the channel at the output of the phase-correcting whitening filter is minimum phase.

Under the ZF criterion, the coefficients of the FBF are equal to the coefficients of the impulse response of $\tilde{g}$ (channel at the output of the FFF) for complete elimination of the post-cursor ISI. We can see this by comparing Fig. 3.10(b) and Fig. 3.10(d), which differ only by the first coefficient.

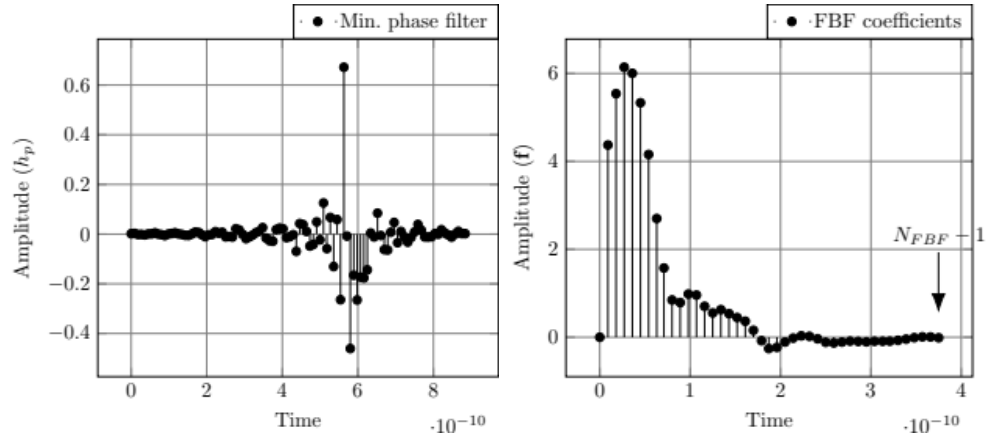
3.4.2.2. DFE with FFF Phase-correcting All-pass Filter

For comparison purposes, we implement the receiver without a matched filter at the receiver. In this case, the FFF is the phase-correcting all-pass filter with impulse response h_p. Referring to Fig. 3.9, we have $h_{rp} = h_p$. The output channel of the FFF is minimum phase but has the same frequency response as the input channel. Note that the impulse response of the FFF is a truncated version to which a delay, denoted δ_{FFF}, is added because the stable impulse response of the phase-correcting all-pass filter is infinite in

duration and non-causal when g (the impulse response of the FFF input channel) is not already at minimum phase.



(a) Matched filter output $h_e(\tau) * h_r$. (b) 4-PAM minimum phase channel $\tilde{g}$.

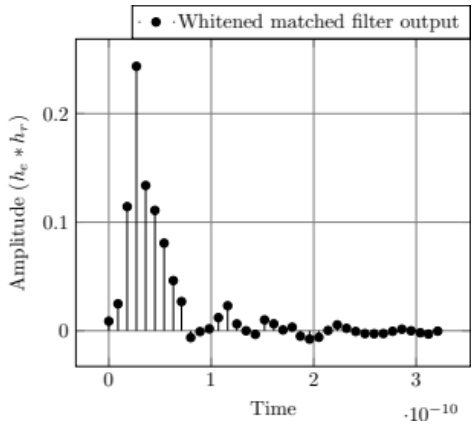


(c) FFF h_p (d) FBF f

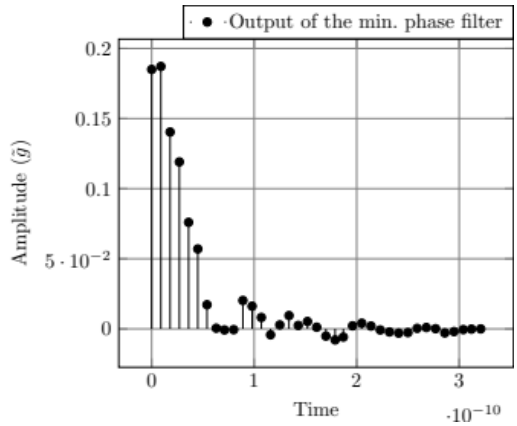
Fig. 3.10. I.R. of the channel at the output of the whitened matched filter (*10.1*), of the minimum phase channel (*10.2*), of the FFF (*10.3*) and of the FBF (*10.4*) for a 4-PAM and considering the whitening matched FFF with $N_{FFF} = 100$.

In the rest of this document, a receiver without a matched filter will have a phase-correcting all-pass filter.

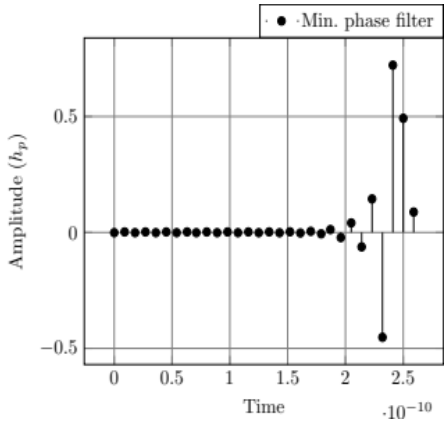
The impulse responses of the FFF and the channel at different points in the receiver without a matched filter are illustrated in Fig. 3.11 at the output of the Mellitz channel already introduced [8].



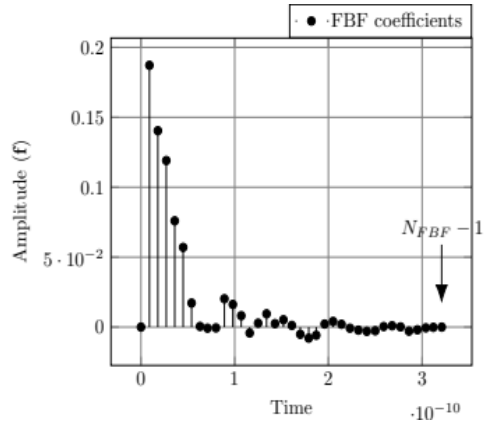
(a) Whitened matched filter output $h_e(\tau) * h_r$.



(b) 4-PAM minimum phase channel $\tilde{g}$.



(c) FFF h_p



(d) FBF f

Fig. 3.11. Global Tx channel (a), minimum phase channel (b), FFF (c) and FBF (d) for a 4-PAM and considering a FFF without a matched filter such that $N_p = 30$.

3.4.2.3. Performance Comparison

We compare, in Fig. 3.12, the performance of the two communication systems.

The number of coefficients in the FFF and FBF differs for the two systems considered because they depend on the shape and size of the impulse response of the overall channel at the DFE input. The FFF must have more coefficients when a matched filter is present in order to correctly minimize the channel phase. If the impulse response of the overall channel at the output of the FFF has $\nu + 1$ coefficients, the FBF has ν taps.

The performance of the “error-free” DFE, corresponding to the DFE without error, is plotted for comparison. The “error-free” DFE differs from the traditional DFE in the nature of the symbols at the input of the FBF. Whereas in the traditional DFE, the symbols

at the input of the FBF are the decisions made by the DFE, in the "error-free" DFE, they are the symbols actually sent. The equation (4) becomes:

$$z_k = \sum_{n=0}^{N_{fbf}-1} f_n a_{k-n+1}$$

The performance of the "error-free" DFE can be simulated to provide a benchmark, but the system cannot be implemented in practice because the symbols sent are unknown to the receiver.

Finally, the performance of a synchronous linear ZF filter in the absence of the matched filter is also plotted to demonstrate the superiority of the DFE over linear filtering in this case [38].

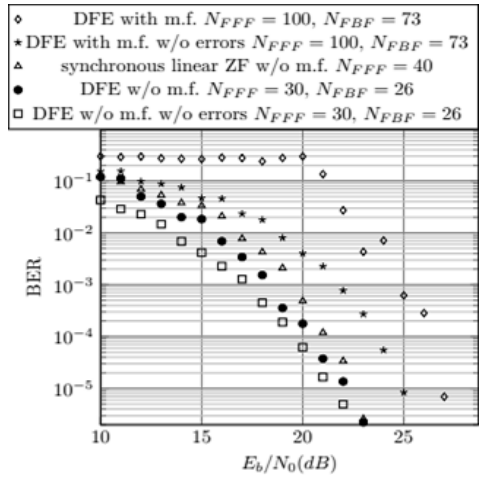


Fig. 3.12. Performance comparison of the communication systems, with and without matched filter (m.f.), for a 4-PAM.

We observe in Fig. 3.12 the superiority of the receiver without a matched filter with an all-pass phase correction filter. The E_b/N_0 gap is approximately of 5 dB for a BER of 10^{-5} between the two DFE. We also note the impact of error propagation in the DFE in the presence of a matched filter. This very strong error propagation is due to the shape of the impulse response of the overall channel at the output of the FFF, illustrated in Fig. 3.10(b), which is very different from the ideal strictly decreasing minimum phase impulse response. The impulse response shown in Fig. 3.11(b) for a receiver without a matched filter is closer to the ideal shape, which results in a small performance difference between a traditional DFE and an "error-free" DFE.

To demonstrate the need to obtain the minimum phase channel at the output of the FFF, we illustrate in Fig. 3.13 the performance of the communication system when the impulse

response at the input of the DFE is the impulse response of the overall channel at transmission, illustrated in Fig. 3.11(a).

In this case, the impulse response of the FFF is the Kronecker delta function and the FBF is the digital impulse response of the overall transmission channel (except for the index coefficient 0). We can see in Fig. 3.13 that the performance of the communication system only improves for values of E_b/N_0 greater than 41 dB. Below this, we observe the effects of very probable error propagation on performance.

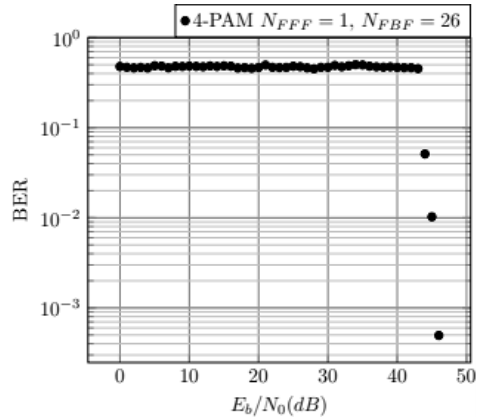


Fig. 3.13. Communication system performance with a DFE without matched filter but with an all-pass phase correcting filter $N_{FFF} = 1$ and $N_{FBF} = 26$.

When shortening the channel upstream of an MLSE block, only partial equalization is necessary, which allows us to use only reduced-amplitude FBF coefficients and thus reduce the magnitude of error propagation. The idea of shortening the channel upstream of an MLSE using a DFE was first presented in [23] and is now considered in certain equalization methods for very high-speed serial links [29].

3.4.3. Structure and Principle of the Partial DFE

To distinguish between DFE for channel shortening and DFE for channel equalization, which we have just introduced, we refer to the former as partial DFE, equipped with partial FBF whose impulse response is denoted by f', and the latter as complete DFE, equipped with complete FBF whose impulse response is denoted by f. We illustrate the partial DFE in Fig. 3.14.

For a given channel, the FFF is the same for the full DFE as for the partial DFE. The vector f′ containing the coefficients of the impulse response of the partial FBF can be expressed in terms of the elements of the vector f containing the coefficients of the full FBF as:

$$\mathrm{f}' = \{0, \ldots, 0, f_{\nu+1}, \ldots, f_{N_{FBF}-1}\}^T,$$

for a channel shortened to $\nu + 1$ coefficients and a full FBF with N_{FBF} coefficients.

We illustrate the coefficients of the partial FBF and the impulse response of the equivalent channel at the DFE output for the receiver with a matched filter in Fig. 3.15 and Fig. 3.16 for the receiver without a matched filter.

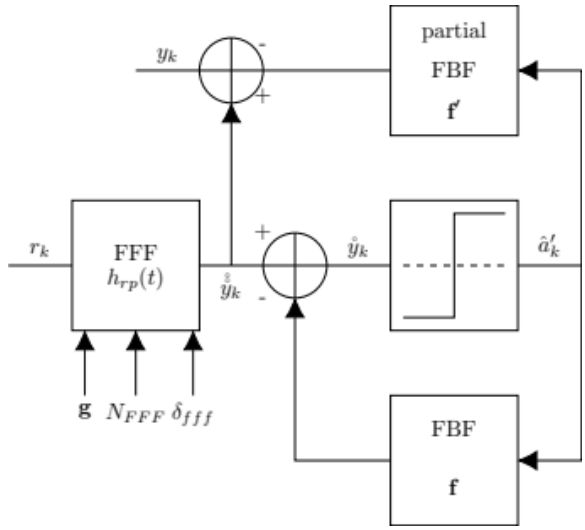
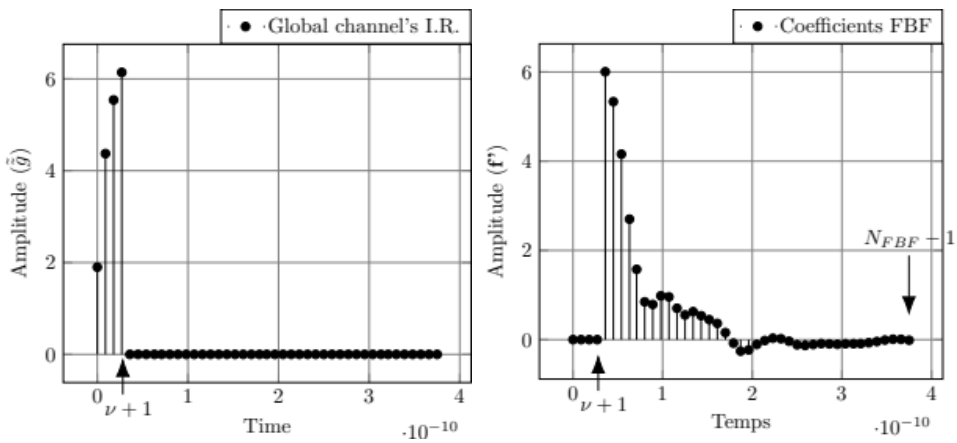
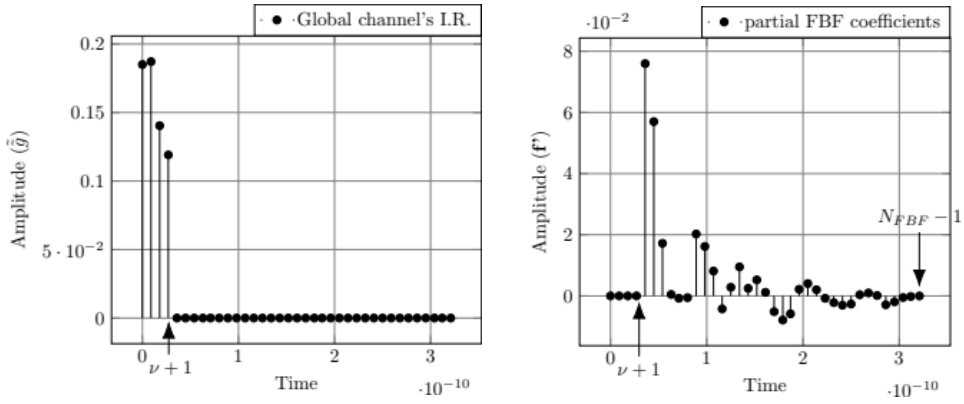


Fig. 3.14. Block diagram of the partial DFE. The inputs to the partial FBF are the outputs of a conventional DFE (with a full FBF). The samples y_k are the transmitted symbols that have passed through the partially equalized channel. They are *the inputs of* the MLSE.



(a) Corresponding global channel I.R. (b) Partial FBF coefficients.

Fig. 3.15. Impulse response of the corresponding global channel after partial DFE and partial FBF coefficients for a receiver with matched filter for a 4-PAM, $N_{FFF} = 100$, $N_{FBF} = 73$ and $\nu + 1 = 4$.



(a) Corresponding global channel I.R. (b) Partial FBF coefficients.

Fig. 3.16. Impulse response of the corresponding global channel after partial DFE and partial FBF coefficients for a receiver without matched filter (replaced by an all-pass phase correcting filter) for a 4-PAM, $N_{FFF} = 30$, $N_{FBF} = 26$ and $\nu + 1 = 4$.

In both cases, since the partial FBF coefficients have smaller amplitudes and are fewer in number than the complete FBF coefficients, we can expect the error propagation due to the DFE feedback loop to be reduced compared to the complete DFE.

Below, we describe the equalization block combining MLSE and DFE for channel shortening and list its advantages and disadvantages.

3.5. Decision Feedback Equalization Followed by a Maximum Likelihood Sequence Estimator – DFE-MLSE

The communication system studied is illustrated in Fig. 3.17. We reuse the complete DFE and partial DFE that we introduced in the previous section. Thus, the FFF can be broken down into an analog prefilter matched to the overall transmission channel followed by a transverse filter, or it can consist exclusively of a transverse filter. Both cases are studied, and the impulse response with and without matched filter is illustrated in Figs. 3.10 and 3.11, respectively, for a global channel impulse response at the MLSE input of size $\nu + 1 = 4$. The output signal of the FFF, denoted $\mathring{\tilde{y}}$, passes through a complete DFE on one side and a partial DFE on the other.

The coefficients of the FBF of the complete DFE are stored in the vector $f = \{\tilde{\tilde{g}}_1, \tilde{\tilde{g}}_2, \ldots, \tilde{\tilde{g}}_{N_{FBF}}\}$ and the decision made by the complete DFE is denoted by $\hat{a}'_k$. We have:

$$\mathring{y}_k = \mathring{\tilde{y}}_k - \sum_{n=0}^{N_{FBF}-1} f_n \hat{a}'_{k-n-1}$$

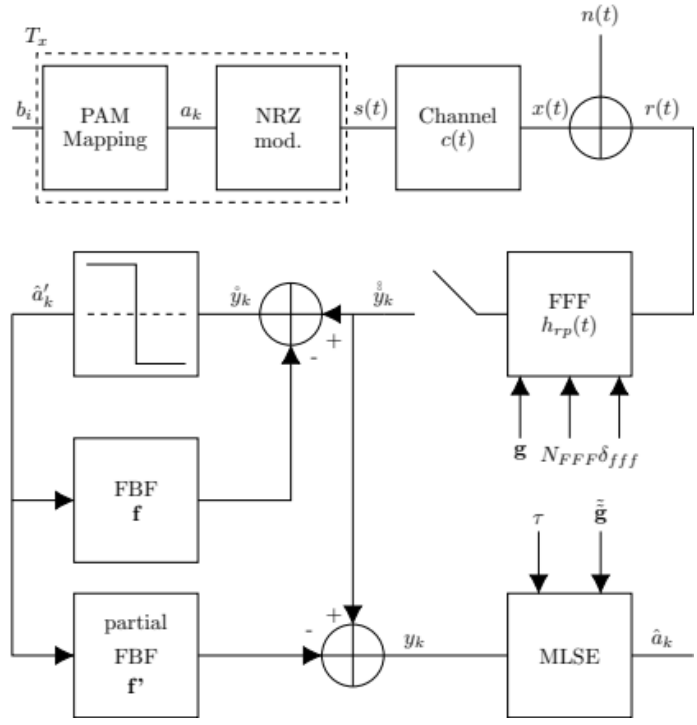


Fig. 3.17. DFE-MLSE: Channel shortening using the partial FBF followed by the MLSE. Decisions at the input of the partial FBF are made by a conventional FBF.

The decisions $\hat{a}'_k$ then pass into the partial FBF, whose $N_{FBF'}$ coefficients are stored in f' before being subtracted from the signal $\mathring{y}$ at the output of the FFF to partially eliminate the ISI. We have:

$$y_k = \mathring{y}_k - \sum_{n=0}^{N_{FBF'}-1} f'_n \hat{a}'_{k-(\nu+1)-n}$$

$$y_k = \mathring{y}_k - \sum_{n=\nu+1}^{N_{FBF}-1} f_n \hat{a}'_{k-n}$$

$$y_k = \mathring{y}_k - \sum_{n=\nu+1}^{N_{FBF}-1} \tilde{\tilde{g}}_{n-1} \hat{a}'_{k-n}$$

With $\nu + 1$ being the size of the partially equalized channel at the input of the MLSE block.

Finally, the partially equalized signal y enters the MLSE, whose state machine and weights associated with its transitions are functions of the TIR denoted by $\tilde{\tilde{g}}$ and the number of non-zero coefficients of the TIR denoted by $\nu + 1$.

The size $\nu + 1$ of the impulse response of the overall channel upstream of the MLSE is determined by computational complexity constraints. The VA makes reliable decisions on the partially equalized signal, reducing the probability of error compared to a simple DFE. Propagation error is limited because the accuracy of the ISI calculation is less decisive in the final decision. Thus, a decision error has less impact on the decision on subsequent symbols.

We observe in Fig. 3.18 a performance gain of more than 3 dB for a BER of 10^{-5} enabled by the addition of an MLSE block at the output of the partial DFE for $\nu + 1 = 4$. When we compare this to Fig. 3.12, we note that the DFE-MLSE receiver achieves better performance than the error-free DFE. Also, the difference between the conventional DFE-MLSE and the error-free DFE-MLSE is less than 1 dB for a BER of approximately 10^{-5}, which is comparable, when looking at Fig. 3.12, to the performance difference between the conventional DFE and the error-free DFE.

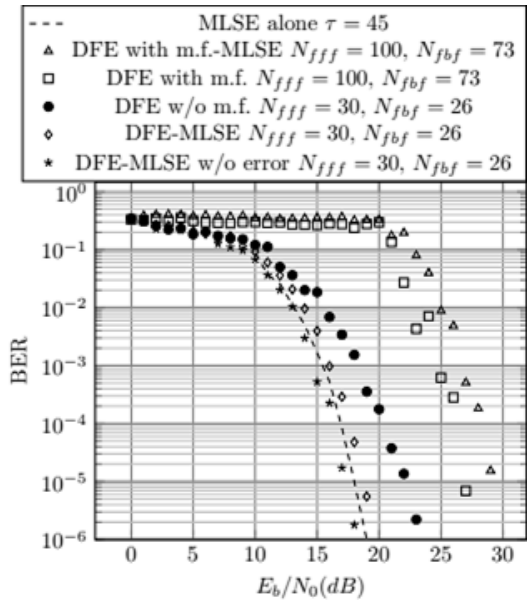


Fig. 3.18. BER comparison of communication systems for a 4-PAM and $\nu + 1 = 4$. Performances of the MLSE alone are also plotted for $\nu + 1 = 10$.

When there is a matched filter at the receiver input, performance is significantly degraded by error propagation, regardless of whether or not there is an MLSE block. Fig. 3.10 clearly shows that the partial FBF coefficients are conducive to error propagation because they have amplitudes comparable to the amplitudes of the coefficients of $\tilde{\tilde{g}}$.

We plotted the performance of the VA alone, for a backtracking depth $\tau = 45$, without pre-filtering at the channel output, and observed that it is (very slightly) better than that of the DFE-MLSE. This observation was expected because the VA is the optimal receiver. To limit the duration of simulations, the impulse response that the VA considers at its

input is reduced to $\nu + 1 = 10$ coefficients, compared to ≈ 40 coefficients for the complete impulse response illustrated in Fig. 3.18, which reduces the performance of the VA. If we were not constrained by the execution time of the algorithm, we would observe that the performance of the VA alone is at least as good as the performance of the DFE-MLSE without error.

To place this study in the context of very high-speed serial links, we note that, with or without a matched filter, the impulse response of the overall channel at the output of the FFF filter has many non-zero coefficients. However, in the specific case of very high-speed serial links, feedback control is difficult to implement because the passage through the FBF must last less than the symbol period and the occupancy of the FBF on the chip must be minimal. According to the literature on the subject, these temporal and spatial constraints allow the DFE's FBF to have only one or two coefficients [30]. To circumvent this limitation, techniques such as sliding block DFE [31] and loop break DFE [32] have been proposed in the literature on very high-speed serial links, but are out of scope of this paper.

3.6. Combination of DFE and MLSE – MLDFE

Finally, the last receiver we consider is the MLDFE, because better performance can be achieved by using an MLSE at the output of a partial DFE, at the cost of greater complexity. Since the decision made by the VA is more reliable than the decision made by the DFE threshold function, Gu and Le Ngoc propose in [33] to feed the decisions made by the VA back into the DFE. The receiver is illustrated in Fig. 3.19.

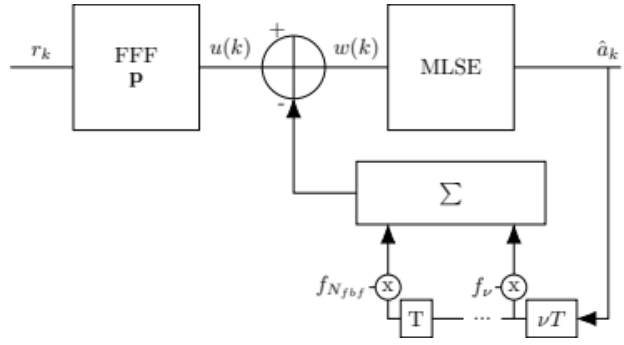


Fig. 3.19. Combination between DFE and MLSE. Decision made by the MLSE is sent back to the input of the partial FBF.

We consider a sequence of L transmitted symbols and a MLSE tracking depth of τ states such that $L > \tau$.

In the case of the DFE-MLSE considered in the previous section, the MLSE processed a sequence of L samples after passing through the partial DFE and estimated a sequence of L symbols.

In this section, for the MLDFE, the MLSE processes a sequence of τ samples at each iteration, estimates a sequence of τ symbols, and delivers only the oldest symbol in this sequence as the final decision. The complete sequence of τ symbols is sent to the FBF to calculate the next τ samples. The sample sequences y_k at the MLSE input therefore overlap on $\tau - 1$ indices but are not necessarily identical because the VA has provided a new sequence estimate.

For each sequence of τ symbols entering the MLSE, the trellis is opened using a preamble consisting of the previous ν final decisions.

We translate this into equations. The sequence of τ samples at the MLSE input can be expressed as:

$$\mathbf{y} = \mathbf{Ap} - \mathbf{Bf'},$$

where y is a column vector of size $\tau \times 1$, A is a matrix $\tau \times N_{fff}$, **p** is a column vector $N_{fff} \times 1$ containing the FFF coefficients, B is a matrix $\tau \times [N_{fbf} - (\nu))$ and **f'** is the column vector $[N_{fbf} - \nu] \times 1$ containing the FBF coefficients. A contains the channel output observations:

$$\mathbf{A} = \begin{pmatrix} r_k & r_{k+1} & \cdots & r_{k+N_{fff}-1} \\ r_{k+1} & r_{k+2} & \cdots & r_{k+N_{fff}-2} \\ \cdots & \cdots & \cdots & \cdots \\ r_{k+\tau-1} & r_{k+\tau} & \cdots & r_{k+\tau+N_{fff}-1} \end{pmatrix}$$

The sequence estimated by the VA is denoted by $\mathbf{o} = \{\hat{a}_k, o_1, \cdots, o_{\tau-1}\}$ with $\hat{a}_k$ being the final decision and o_i being the other estimated symbols in the sequence. **B** is the matrix consisting of estimates denoted by o_i with $i \in [\![0, \tau - 1]\!]$ when they are temporary, and denoted by $\hat{a}_{k-i}$ with $i \in [\![1, N_{fbf}]\!]$, when the decision is final. Thus[1]:

$$\mathrm{B} = \begin{pmatrix} \hat{a}_{k-\nu} & \hat{a}_{k-\nu-1} & \cdots & \hat{a}_{k-N_{FBF}} \\ \hat{a}_{k-\nu+1} & \hat{a}_{k-\nu} & \cdots & \hat{a}_{k-N_{FBF}+1} \\ \cdots & \cdots & \cdots & \cdots \\ \hat{a}_k & \hat{a}_{k-1} & \cdots & \hat{a}_{k-N_{FBF}+\nu} \\ o_1 & \hat{a}_k & \cdots & \hat{a}_{k-N_{FBF}+\nu+1} \\ \cdots & \cdots & \cdots & \cdots \\ o_{\tau-1-\nu} & o_{\tau-\nu-2} & \cdots & o_{\tau-\nu-N_{FBF}} \end{pmatrix}$$

In the following section, we illustrate the performance of this receiver and compare it to the performance of other previously introduced receivers.

[1] Note that $o_{\tau-\nu-N_{FBF}}$ may not exist depending on the system parameters. If $\tau > N_{FBF} + \nu$, it exists. Otherwise, this element is 0. Given the orders of magnitude of N_{FBF} relative to τ, this borderline case is anecdotal. Also, in matrix B, we note that the element in the upper left corner is not $\hat{a}_k$ because the partial DFE only shortens the channel.

3.7. Performance Comparison through Simulation

As already mentioned above, the performance of a nonlinear equalizer is difficult to describe analytically. It is possible to frame the performance with bounds in the case of MLSE or to give an upper limit of performance by neglecting error propagation in the case of DFE, but to get a precise idea of the performance of communication systems equipped with different receivers, we must resort to simulations. We therefore propose to compare the performance of the different nonlinear equalization methods presented using simulations. We first compare the performance of the receivers without constraining the complexity and then we compare the performance taking into account the limitations set by the standard.

3.7.1. Performance Comparison without Complexity Constraints

We consider the channels, already presented above, proposed by R. Mellitz to the IEEE 802.3df working group [8]. The characteristics of these channels have already been illustrated in Fig. 3.6. These characteristics are the result of experiments and physical models to determine the frequency attenuations due to passage through the substrate and impedance differences.

Given that, for finite filter sizes and the frequency-selective channels studied, the communication system with a matched filter performs less well than the communication system without a matched filter, in this section we are only interested in the communication system without a matched filter at the receiver. We also consider that the receiver is equipped with a filter that provides a minimum phase global channel at the input of the channel shortening pre-equalizer. The receiver of the communication system under consideration is illustrated in Fig. 3.20 for the Falconer-MLSE and the DFE-MLSE. The "channel shortening pre-equalizer" block can either be a Falconer filter or a channel shortening DFE.

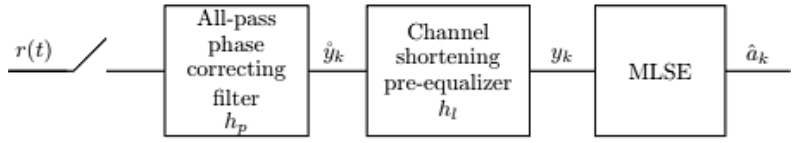


Fig. 3.20. Receiver block scheme with channel shortening pre-equalizer.

The channel phase correction passband filter has $N_p = 10$ coefficients, the minimum size required to correctly render the phase of the channel considered minimal in 4-PAM.

For channel shortening using Falconer's linear filter, the number of coefficients that make up the filter is determined experimentally. We have determined that the performance of the Falconer-MLSE communication system is best for $N_{FM} = 71$.

In practice, the two digital filters can be combined to form a single filter.

The performance of the receivers is illustrated in Fig. 3.21 for a partially equalized global channel impulse response of $\nu + 1 = 4$ coefficients and a 4-PAM, initially.

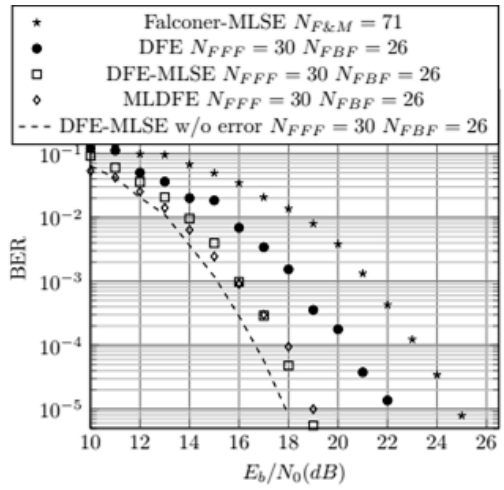


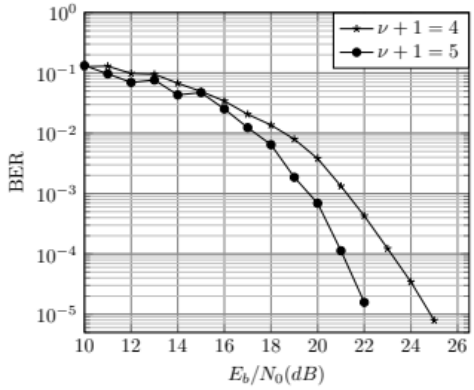
Fig. 3.21. Performance comparison between non-linear equalization algorithms for the channel proposed by Mellitz, for a 4-PAM *and for* $\nu + 1 = 4$.

We are interested in performance differences between receivers for a BER of 10^{-5}. We observe that the method of channel shortening using the Falconer filter followed by MLSE is the least efficient communication system among those studied. The use of DFE alone provides a gain of 3 dB. Channel shortening using DFE followed by estimation using the Viterbi algorithm provides a gain of 3 dB compared to DFE alone. Finally, the performance gap between MLDFE and DFE-MLSE is less than 0.5 dB in favor of DFE-MLSE.

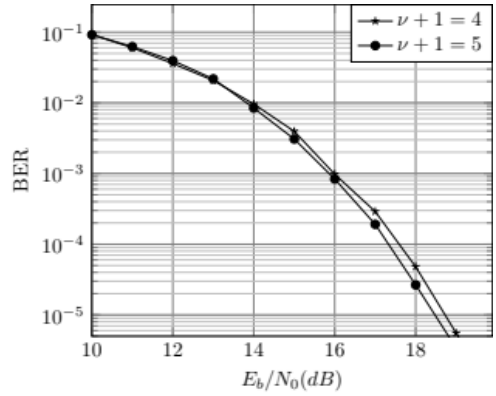
To assess the influence of the size of the impulse response of the MLSE input channel on the relative performance of the receivers, we perform the same simulations for $\nu + 1 = 5$. Greater computing power is allocated to MLSE.

Channel shortening using a transverse filter is always the least effective method. However, the performance gap between DFE alone and Falconer filter followed by MLSE is reduced to less than 2 dB for a BER of 10^{-5} after comparison with Fig. 3.22. The performance of DFE followed by MLSE is significantly improved by 0.4 dB, and the performance of MLDFE (or MLDFE without error, not shown here) is improved by 1 dB. MLDFE performs better than DFE-MLSE in this case.

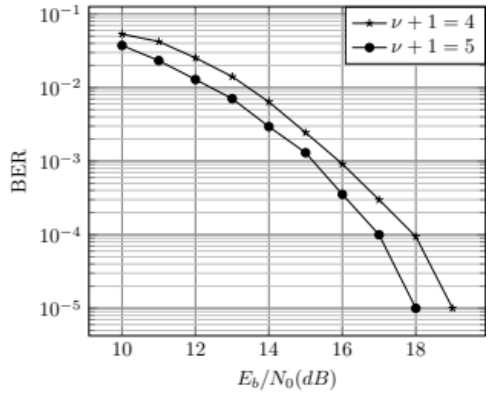
This comparison shows the benefits of increasing the size of the shortened channel impulse response for Falconer-MLSE, DFE-MLSE, and MLDFE. However, computational complexity increases with the increase in the size of the impulse response of the partially equalized channel at the MLSE input. The complexity of the Viterbi algorithm is multiplied by 4 for a 4-PAM when ν increases by 1.



Falconer-MLSE $N_{FM} = 71$.



DFE-MLSE $N_{FFF} = 30, N_{FBF} = 26$.



MLDFE $N_{FFF} = 30, N_{FBF} = 26$.

Fig. 3.22. Performance comparison between non-linear equalization algorithms for the channel proposed by Mellitz for $\nu + 1 = 4$ and $\nu + 1 = 5$.

After illustrating the performance of receivers for 4-PAM in Fig. 3.21, we illustrate their performance for 8-PAM in Fig. 3.23 for a fixed bit rate.

We observe a deterioration in receiver performance involving decision feedback compared to 4-PAM. Only Falconer-MLSE performs identically to 4-PAM. DFE-MLSE corresponds to DFE-MLSE without error. We conclude that for $\nu + 1 = 5$, there is no error propagation in this receiver.

We note that in the case of SerDes links operating at 224 Gbps, the standard deviation of time jitter is of the order of $\sigma_j = 0.01T_b$. That means 1 % of the bit period. We have verified that for such a low level of jitter, its effect is not noticeable on receiver performance. To observe the effect of jitter, its standard deviation would need to be 10 times higher, of the order of $\sigma_j = 0.1T_b$.

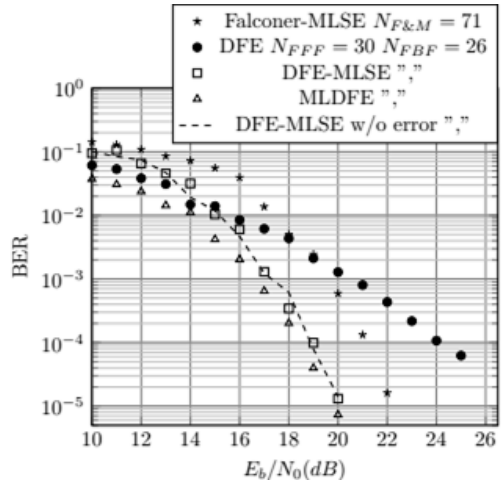


Fig. 3.23. Performance comparison between different receivers for $\nu + 1 = 5$ *and for an* 8-PAM.

3.7.2. Performance Comparison under Standard Constraints

Considering very high data rate serial links, a receiver is fit if it verifies some constraints. It has to reach a BER of 10^{-5}, or lower, for aE_b/N_0equal to 30 dB before decoding according to the standard [40]. The computational complexity has to be kept to its minimum, thus the impulse response of the overall channel at the MLSE input is reduced to $\nu + 1 = 2$, the minimum size for the Viterbi algorithm to perform better than a threshold decision block. Finally, because both occupied area on the chip has to be small and timing delay constraints when using a feedback loop are very tight, the FBF of the channel shortening DFE can only have up to two coefficients [30].

To verify the standard's constraints in terms of computational complexity and performance without coding, we propose a receiver with a Falconer filter, DFE, and MLSE in series, which performance for different parameters is illustrated in Fig. 3.24.

We compare Fig. 3.24, the performance of Falconer-DFE-MLSE with the performance of Falconer-MLSE for a 4-PAM and an 8-PAM. We denote $\nu' + 1$ as the size of the impulse response at the output of the Falconer filter and $\nu + 1$ as the size of the impulse response at the input of the VA. In the case of Falconer-MLSE, we have $\nu = \nu'$. In the case of Falconer-DFE-MLSE, since the full FBF can only have a maximum of two coefficients and $\nu + 1 = 2$, we have $\nu' + 1 = 3$ and therefore a partial FBF with $N_{FBF,} = 1$ coefficient.

For a 4-PAM, we observe that the performance of the Falconer-MLSE receiver is better than the performance of the Falconer-DFE-MLSE by 2 dB for a BER of 10^{-5}.

For 8-PAM, the performance of the Falconer-DFE-MLSE receiver reaches an error floor for an BER of 10^{-5} due to error propagation (as verified experimentally), while the Falconer-MLSE has no error floor.

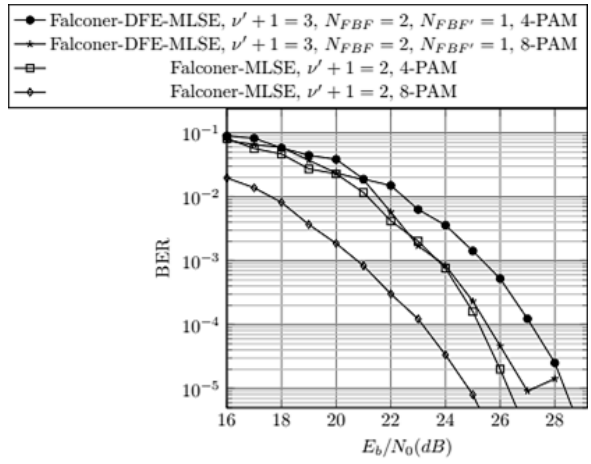


Fig. 3.24. Performance comparison between receivers Falconer-DFE-MLSE with all-pass phase correcting filter $N_p = 10$, Falconer filter $N_{FM} = 71$, DFE depending on the use case $N_{FFF} = 1, N_{FBF} = 2, N_{FBF'} = 1$ and MLSE $\nu + 1 = 2$.

When we allow the Falconer-DFE-MLSE's full FBF to have 3 coefficients, this receiver has a 2 dB gain for a BER of 10^{-5} compared to the Falconer-MLSE, as shown in Fig. 3.25.

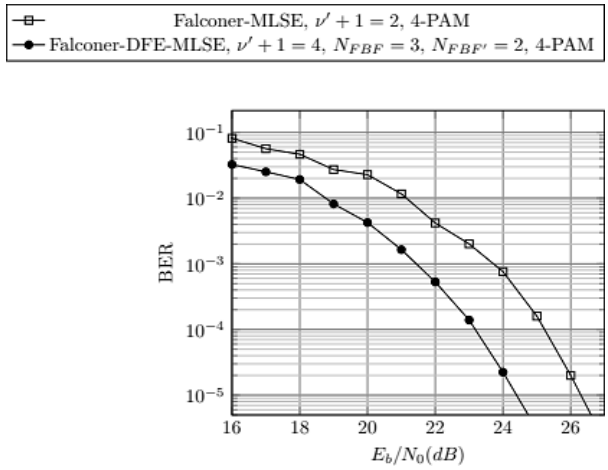


Fig. 3.25. Performance comparison between Falconer-DFE-MLSE *with* $\nu' + 1 = 4, N_{FBF} = 3, N_{FBF'} = 2$ *and* Falconer-MLSE.

We conclude that Falconer-DFE-MLSE remains interesting if we disregard the spatial occupancy constraint that limits the number of FBF coefficients.

Since the standard specifications require a BER of less than 10^{-5} for a $E_b/N_0 = 30\ dB$ ratio [40], we conclude that for a 4-PAM, the proposed Falconer-DFE-MLSE receiver meets the specifications for limited computational complexity and a FBF coefficient number less than 2 (Fig. 3.24). For 8-PAM, with performance reaching an error floor for a BER of 10^{-5}, the Falconer-DFE-MLSE receiver is unsuitable. The only receiver with nonlinear equalization that meets the specifications for limited complexity is the Falconer-MLSE.

In parallel with our work, in December 2024, a team published an article [7] providing a very comprehensive state-of-the-art review of equalization methods used in very high-speed serial links. They detail the concrete implementation of the systems from a circuit perspective and compare the proposed nonlinear equalization solutions in terms of BER, energy efficiency, computational complexity, and area occupied. The systems with the most interesting characteristics are those combining linear filtering, DFE, and MLSE based on the Viterbi algorithm [34-36]. We converged on the same type of receiver, but we also found that the structure of the optimal receiver depended on several parameters (channel impulse response shape and modulation order). Our receiver with a linear filtering-DFE-MLSE structure differs from those proposed in the literature in terms of the linear filtering strategy used (phase correction followed by Falconer filter).

3.8. Conclusion

To conclude this chapter, we summarize the observations and conclusions made for each receiver considered in Table 3.1. To be suitable for very high-speed serial links, a receiver must have limited computational complexity, a maximum FBF of two coefficients when a DFE is used to shorten the channel, and achieve a BER of 10^{-5} for an $E_b/N_0 = 30$ dB ratio.

For the channel under consideration, synchronous linear ZF filtering meets the specifications imposed by the standard, as shown by the performance illustrated in Fig. 3.12. MLSE alone is too complex, and DFE alone has an FBF with more than two coefficients to verify the ZF criterion.

Of the first three nonlinear receivers considered, Falconer-MLSE, DFE-MLSE, and MLDFE, only Falconer-MLSE meets all three requirements. In fact, MLDFE is the most efficient receiver studied in terms of BER, but the execution time of the VA makes it incompatible with decision feedback equalization. The DFE-MLSE rivals the MLDFE in the scenario ($\nu + 1 = 4$, 4-PAM) and the scenario ($\nu + 1 = 5$, 8-PAM), but the FBF has more than two coefficients. The Falconer-MLSE, on the other hand, does not require decision feedback and can reduce the size of the channel impulse response to two coefficients.

In order for the DFE-MLSE to meet the FBF size criterion, the channel is first shortened using the Falconer filter. This new combination, called Falconer-DFE-MLSE, performs 2 dB worse in the scenario ($\nu' + 1 = 3, \nu + 1 = 2$, 4-PAM) than the Falconer-MLSE and reaches an error floor for a BER of 10^{-5} in the scenario ($\nu' + 1 = 3, \nu + 1 = 2$, 8-PAM). We note that when we remove the constraint on the maximum number of FBF coefficients, Falconer-DFE-MLSE can perform better than Falconer-MLSE, as in the scenario ($\nu' + 1 = 4, \nu + 1 = 2$, 4-PAM), for example.

Table 3.1. Summary table of the advantages and disadvantages of the different receivers studied. Receivers whose names are in green are those likely to meet the standard's specifications.

Receiver	Advantages	Disadvantages
DFE alone	Limited complexity and thermal noise amplification	Error propagation
MLSE alone	Optimal performance	High complexity
Linear filtering alone	Limited complexity, space requirements, and power consumption	Thermal noise amplification
Falconer-MLSE	Limited complexity	Amplification of thermal noise
DFE-MLSE	Limited amplification of thermal noise and error propagation	FBF too long
MLDFE	Same as above	Complexity too high
Falconer-DFE-MLSE	Limited complexity	Thermal noise amplification and error propagation

The three receivers we have therefore selected for equalizing very high-speed SerDes links are synchronous linear ZF filtering, Falconer-MLSE, and Falconer-DFE-MLSE. Their name is written in green in Table 3.1. Although the synchronous ZF filter has the lowest computational complexity, in cases where the channel spectrum has one or more zeros (very strong attenuation), linear filtering alone cannot adequately equalize the channel, justifying the use of VA. Furthermore, Falconer-DFE-MLSE remains interesting when more degrees of freedom are granted to the system, which is conceivable with future technological advances. The three receivers can therefore be considered according to the channel and the modulation order.

References

[1]. C. A. Belfiore, J. H. Park, Decision feedback equalization, *Proceedings of the IEEE*, Vol. 67, Issue 8, 1979, pp. 1143-1156.

[2]. A. J. Viterbi, Error bounds for convolutional codes and an asymptotically optimum decoding algorithm, *IEEE Transactions on Information Theory*, Vol. 13, Issue 2, 1967, pp. 260-269.

[3]. A. J. Viterbi, Convolutional codes and their performance in communication systems, *IEEE Transactions on Communication Technology*, Vol. 19, Issue 5, 1971, pp. 751-772.

[4]. G. D. Forney, Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference, *IEEE Transactions on Information Theory*, Vol. 18, Issue 3, 1972, pp. 363-378.
[5]. J. K. Omura, Optimal receiver design for convolutional codes and channels with memory via control theoretical concepts, *Information Sciences*, Vol. 3, Issue 3, 1971, pp. 243-266.
[6]. H. Kobayashi, Application of probabilistic decoding to digital magnetic recording systems, *IBM Journal of Research and Development*, Vol. 15, Issue 1, 1971, pp. 64-74.
[7]. S. Jang, J. Lee, Y. Choi, D. Kim, et al., Recent advances in ultra-high-speed wireline receivers with ADC-DSP-based equalizers, *IEEE Open Journal of the Solid-State Circuits Society*, Vol. 4, 2024, pp. 29-45.
[8]. R. Mellitz, A collection of cabled backplane prototype channels for 200 Gbps per lane for .3df PHY type development, IEEE P802.3df & IEEE P802.3dj Tools and Channel Data Area, 2022, https://www.ieee802.org/3/df/public/adhoc/electrical/22_0502/mellitz_3df_elec_01_220502.pdf
[9]. H. Kobayashi, Correlative level coding and maximum-likelihood decoding, *IEEE Transactions on Information Theory*, Vol. 17, Issue 5, 1971, pp. 586-594.
[10]. F. Magee, J. Proakis, Adaptive maximum-likelihood sequence estimation for digital signaling in the presence of intersymbol interference, *IEEE Transactions on Information Theory*, Vol. 19, Issue 1, 1973, pp. 120-124.
[11]. S. Qureshi, E. Newhall, Adaptive receiver for data transmission over time-dispersive channels, *IEEE Transactions on Information Theory*, Vol. 19, Issue 4, 1973, pp. 448-457.
[12]. G. D. Forney, The Viterbi algorithm, *Proceedings of the IEEE*, Vol. 61, Issue 3, 1973, pp. 268-278.
[13]. T. Ericson, Structure of optimum receiving filters in data transmission systems, *IEEE Transactions on Information Theory*, Vol. 17, Issue 3, 1971, pp. 352-353.
[14]. J.-M. Brossier, Signal et Communication Numérique: Égalisation et Synchronisation, *Hermès Science*, 1997.
[15]. R. W. Lucky, Automatic equalization for digital communication, *Bell System Technical Journal*, Vol. 44, Issue 4, 1965, pp. 547-588.
[16]. T. Klein, J. Wolf, On the use of channel introduced redundancy for error correction, *IEEE Transactions on Communication Technology*, Vol. 19, Issue 4, 1971, pp. 396-402.
[17]. D. D. Falconer, F. Magee Jr, Adaptive channel memory truncation for maximum likelihood sequence estimation, *Bell System Technical Journal*, Vol. 52, Issue 9, 1973, pp. 1541-1562.
[18]. N. Al-Dhahir, J. M. Cioffi, Efficiently computed reduced-parameter input-aided MMSE equalizers for ML detection: A unified approach, *IEEE Transactions on Information Theory*, Vol. 42, Issue 3, 1996, pp. 903-915.
[19]. C. Beare, The choice of the desired impulse response in combined linear-Viterbi algorithm equalizers, *IEEE Transactions on Communications*, Vol. 26, Issue 8, 1978, pp. 1301-1307.
[20]. D. G. Messerschmitt, A geometric theory of intersymbol interference: Part I: Zero-forcing and decision-feedback equalization, *Bell System Technical Journal*, Vol. 52, Issue 9, 1973, pp. 1483-1519.
[21]. M. E. Austin, Decision-feedback equalization for digital communication over dispersive channels, PhD Thesis, *Massachusetts Institute of Technology*, Cambridge, 1967.
[22]. P. Monsen, Feedback equalization for fading dispersive channels, *IEEE Transactions on Information Theory*, Vol. 17, Issue 1, 1971, pp. 56-64.
[23]. W. Lee, F. Hill, A maximum-likelihood sequence estimator with decision-feedback equalization, *IEEE Transactions on Communications*, Vol. 25, Issue 9, 1977, pp. 971-979.
[24]. J. Labat, O. Macchi, C. Laot, Adaptive decision feedback equalization: Can you skip the training period?, *IEEE Transactions on Communications*, Vol. 46, Issue 7, 1998, pp. 921-930.

[25]. R. Price, Nonlinearly feedback-equalized PAM vs. capacity, in *Proceedings of the IEEE International Conferences on Communications*, 1972, pp. 22-12 - 22-17.

[26]. J. Salz, Optimum mean-square decision feedback equalization, *Bell System Technical Journal*, Vol. 52, Issue 8, 1973, pp. 1341-1373.

[27]. E. Shamash, K. Yao, On the structure and performance of a linear decision feedback equalizer based on the minimum error probability criterion, in *Proceedings of the 10th International Conference on Communications*, 1974, pp. 25F1-25F5.

[28]. J. M. Cioffi, G. P. Dudevoir, M. V. Eyuboglu, G. D. Forney, MMSE decision-feedback equalizers and coding – Part I: Equalization results, *IEEE Transactions on Communications*, Vol. 43, Issue 10, 1995, pp. 2582-2594.

[29]. M. E. Meybodi, H. Gomez, Y.-C. Lu, H. Shakiba, et al., Design and implementation of an on-demand maximum-likelihood sequence estimation (MLSE), *IEEE Open Journal of Circuits and Systems*, Vol. 3, 2022, pp. 97-108.

[30]. M.-A. LaCroix, H. Wong, Y. H. Liu, H. Ho, et al., A 60 Gb/s PAM-4 ADC-DSP transceiver in 7 nm CMOS with SNR-based adaptive power scaling achieving 6.9 pJ/b at 32 dB loss, in *Proceedings of the IEEE International Solid-State Circuits Conference (ISSCC'19)*, 2019, pp. 114-116.

[31]. J. Bailey, H. Shakiba, E. Nir, G. Marderfeld, et al., A 112-Gb/s PAM-4 low-power nine-tap sliding-block DFE in a 7-nm FinFET wireline receiver, *IEEE Journal of Solid-State Circuits*, Vol. 57, Issue 1, 2022, pp. 32-43.

[32]. D. Kim, Y. Choi, J. Lee, S. Jang, et al., A loop-break decision feedback equalizer for DAC/ADC-DSP-based wireline transceivers, *IEEE Transactions on Circuits and Systems I: Regular Papers*, Vol. 71, Issue 8, 2024, pp. 3674-3684.

[33]. Y. Gu, T. Le-Ngoc, Adaptive combined DFE/MLSE techniques for ISI channels, *IEEE Transactions on Communications*, Vol. 44, Issue 7, 1996, pp. 847-857.

[34]. D. Pfaff, M. Nummer, N. Hai, P. Xia, et al., A 224 Gb/s 3 pJ/b 40 dB insertion loss transceiver in 3 nm FinFET CMOS, in *Proceedings of the IEEE International Solid-State Circuits Conference (ISSCC'24)*, 2024, pp. 128-130.

[35]. S. Song, K. D. Choo, T. Chen, S. Jang, et al., A maximum-likelihood sequence detection powered ADC-based serial link, *IEEE Transactions on Circuits and Systems I: Regular Papers*, Vol. 65, Issue 7, 2018, pp. 2269-2278.

[36]. H. Yueksel, M. Braendli, A. Burg, G. Cherubini, et al., Design techniques for high-speed multi-level Viterbi detectors and trellis-coded-modulation decoders, *IEEE Transactions on Circuits and Systems I: Regular Papers*, Vol. 65, Issue 10, 2018, pp. 3529-3542.

[37]. P. Miqueu, F. Belveze, J. M. Brossier, L. Ros, Non-linear equalization techniques for high data rates serial links, in *Proceedings of the 5th IFSA Winter Conference on Automation, Robotics & Communications for Industry 4.0/5.0 (ARCI'25)*, Granada, Spain, 19-21 Feb. 2025, pp. 154-159.

[38]. P. Miqueu, F. Belveze, J. M. Brossier, L. Ros, Parameters setting based on bit error probability formula for an amplitude modulated, baseband, communication system affected by timing jitter and frequency selective channel, *AEU – International Journal of Electronics and Communications*, Vol. 178, 2024, 155288.

[39]. IEEE P802.3df & IEEE P802.3dj Tools and Channel Data Area, https://www.ieee802.org/3/df/public/tools/index.html

[40]. X. Wang, IEEE P802.3df 200 Gb/s, 400 Gb/s, 800 Gb/s, and 1.6 Tb/s Ethernet Task Force Architecture and Logic Ad Hoc Area, 2022, https://www.ieee802.org/3/df/public/adhoc/logic/22_0411/wang_3df_logic_220411.pdf

Advances in Networks, Security and Communications, Volume 4

Sergey Y. Yurish, Editor

Advances in Networks, Security and Communications, Vol. 4 is a practical, research-driven guide to a single challenge shared by today's digital infrastructure. This volume brings together three complementary perspectives—protocol design, resilient positioning, and receiver processing—to show what "robustness" looks like in practice.

The book opens with a safety-focused communication view: a grey-channel approach that treats the channel as measurable and manageable rather than perfectly trusted or completely unknown. It then turns to vehicle localization beyond GNSS, demonstrating how MEMS sensor data and map-based reasoning can support positioning when satellite navigation is unreliable. Finally, it addresses the physical layer, presenting high-data-rate equalization techniques—including channel shortening and Viterbi/MLSE-based methods—to mitigate intersymbol interference in frequency-selective channels.

Written for graduate students, researchers, and practicing engineers, this volume is especially relevant to professionals in industrial automation and safety-critical networking, intelligent transportation and mobility systems, and digital communications and signal processing. Its value lies in its clear problem framing, implementable methods, and engineering-minded trade-offs—making it both a concise reference and a source of ideas for building networks and communication systems that remain trustworthy under real operational constraints.

ISBN 978-84-09-82029-0

9788409820290