electronics

MDPI

Article

Enhancing the Coupling of Real-Virtual Prototypes: A Method for Latency Compensation

Peter Baumann [1], Oliver Kotte [2], Lars Mikelsons [3] and Dieter Schramm [4,*]

[1] Robert Bosch GmbH, Wilhelm-Fein-Straße 6, 7163 Ludwigsburg, Germany; peter.baumann5@de.bosch.com
[2] Robert Bosch GmbH, Robert-Bosch-Straße 2, 71701 Schwieberdingen, Germany
[3] Chair of Mechatronics, University of Augsburg, Am Technologiezentrum 8, 86159 Augsburg, Germany
[4] Chair of Mechatronics, University of Duisburg-Essen, Lotharstraße 1, 47057 Duisburg, Germany
* Correspondence: dieter.schramm@uni-due.de

Abstract: Currently, innovations in mechatronic products often occur at the system level, requiring consideration of component interactions throughout the entire development process. In the earlier phases of development, this is accomplished by coupling virtual prototypes such as simulation models. As the development progresses and real prototypes of certain system components become available, real-virtual prototypes (RVPs) are established with the help of network communication. However, network effects—all of which can be interpreted as latencies in simplified terms—distort the system behavior of RVPs. To reduce these distortions, we propose a coupling method for RVPs that compensates for latencies. We present an easily applicable approach by introducing a generic coupling algorithm based on error space extrapolation. Furthermore, we enable online learning by transforming coupling algorithms into feedforward neural networks. Additionally, we conduct a frequency domain analysis to assess the impact of coupling faults and algorithms on the system behavior of RVPs and derive a method for optimally designing coupling algorithms. To demonstrate the effectiveness of the coupling method, we apply it to a hybrid vehicle that is productively used as an RVP in the industry. We show that the optimally designed and trained coupling algorithm significantly improves the credibility of the RVP.

Keywords: real-virtual prototypes; real-time simulation; co-simulation; hardware-in-the-loop; functional mock-up interface; FMI; DCP; distributed simulation; latency compensation; discontinuity detection

check for updates

Citation: Baumann, P.; Kotte, O.; Mikelsons, L.; Schramm, D. Enhancing the Coupling of Real-Virtual Prototypes: A Method for Latency Compensation. *Electronics* **2024**, *13*, 1077. https://doi.org/10.3390/electronics13061077

Academic Editors: Miro Milanovic, Eric Monmasson, Franc Mihalič and Lahoucine Idkhajine

Received: 9 February 2024
Revised: 11 March 2024
Accepted: 12 March 2024
Published: 14 March 2024

1. Introduction

Many innovations in established and technologically advanced mechatronic products go beyond individual components and, instead, occur at the system level through a holistic approach. This involves combining information from different domains or connecting multiple independent products from different ecosystems, which increases the complexity of the product under development [1]. The automotive industry offers significant potential for innovations at the system level, as modern vehicles interact in an ecosystem with other vehicles, consumer electronics, or energy systems, such as charging infrastructure. However, the successful implementation of cross-domain features, such as advanced driver assistance systems, requires reliable coordination and interaction between various vehicle components, such as driving dynamics, steering, and braking. For the efficient and cost-effective development of cross-domain features, it is crucial to consider the interactions between the different domains already at the early stages of the development process [2]. This is achieved by implementing virtual prototypes in the form of simulation models of the individual components of the mechatronic system and coupling them together to create large model-in-the-loop (MIL) simulation environments using co-simulation methods [3]. In the last decade, the functional mock-up interface (FMI) standard [4], for which version 3.0 has recently been published [5], has established itself as a widely adopted interface format for virtual prototypes. It facilitates the creation of MIL and software-in-the-loop (SIL) simulations by enabling the coupling of virtual prototypes, containerized as functional mock-up units (FMUs), across different simulation tools.

While simplified simulation models may be sufficient for MIL simulations at the start of new product developments, the models become increasingly detailed as the requirements become more precise over the course of product development. Given the significant investment of time and resources in creating virtual prototypes during the earlier stages of the development process, their reusability in all phases of the development process must be ensured, especially as the development cycles of products continue to shorten [6]. During the later stages of development, such as the verification and validation phase, when the first components of the product are already available as real, physical hardware prototypes, it becomes necessary to couple real and virtual prototypes, resulting in what is known as real-virtual prototypes (RVPs) [7]. We use the term RVP instead of hardware-in-the-loop (HIL) [8] or X-in-the-loop (XIL) [9] to emphasize that the virtual prototypes of RVPs are the original soft real-time simulation models implemented during earlier phases of the development process, rather than optimized real-time simulation models specifically designed to be coupled with hardware components, such as test benches.

The mentioned continuity from purely virtual MIL to RVPs is the subject of research in various industries. Starting from a cross-domain MIL simulation, the authors of [10] demonstrate the transition to an RVP by replacing the simulation model of an internal combustion engine with an engine test bench. In [11], an RVP was implemented to validate real relays within smart power grids, and the authors of [7] present information on how RVPs are utilized in the maritime domain for the development of position controllers for container ships [12]. RVPs can also incorporate a human interface, such as coupling detailed simulation models with a driving simulator to assess the effects of various vehicle configurations on the driver [13]. Furthermore, RVPs are often spatially distributed, as multiple real prototypes on different test benches are coupled, which requires communication via a network to exchange signals and enables interactions between the coupled prototypes. For example, in Europe, an RVP containing five test benches is being established for the development of electric vehicles [14,15].

A lot of research has been dedicated to the practical integration and configuration of the coupling between real and virtual prototypes, which has resulted in the development of the distributed co-simulation protocol (DCP) specification [16]. Just as the FMI facilitates the setup of co-simulations, the DCP standardizes the configuration and operation of the network communication between coupled prototypes, which enables the efficient setup of RVPs, even when the coupled prototypes originate from different vendors [17,18]. Since its introduction, the DCP has been improved several times, for example, in [19], where the DCP was combined with an already existing co-simulation interface to increase its compatibility with certain hardware platforms. Amongst others, the DCP has been successfully adopted by the authors of [20] to couple an electric machine and a combustion engine test bench, in [21], for coupling a 3D environment simulation to a vehicle simulation, and in our previous contribution [22] to couple a real and a virtual prototype across a large distance. However, even with the standardized configuration of RVPs through the DCP, not all challenges in terms of their usage are resolved. One remaining challenge lies in the occurrence of coupling faults in the communication between the prototypes due to network effects. The resulting coupling errors can negatively affect the system's behavior, thus constraining the applicability of RVPs. Therefore, there is a need for research to enhance the coupling of RVPs by compensating for coupling faults.

This contribution presents a coupling method that enables increased utilization of RVPs in product development by reducing coupling errors through the compensation of dominant network effects during data exchange between prototypes. The focus is on ensuring the method's simplicity for general application without requiring the modification of the coupled virtual and real prototypes. Instead, the compensation of network effects shall solely rely on coupling signals and directly measurable information within the network.

In order to achieve this, we first introduce real-time co-simulation in Section 2.1, which is the methodological basis for describing the coupling process of RVPs. We identify that network effects leading to coupling faults in RVPs can be interpreted in a simplified manner solely as latencies. Additionally, we define a general form of linear coupling algorithms considered in this contribution to compensate for these latencies. Afterward, in Section 2.2, we present a novel procedure for extrapolation in the error space and derive a new, generic coupling algorithm from it. Notably, the proposed coupling algorithm is generally applicable without requiring knowledge of the RVP's system behavior. This constitutes a significant difference from previous works, where the coupling algorithm relies on precise mathematical models of all coupled prototypes, such as the utilization of a Smith predictor [23], a Luenberger observer [24], or model predictive control methods [25] for latency compensation.

Subsequently, in Section 2.3, we build upon our approach that was previously published in [26] to utilize small neural networks as coupling algorithms. In the present contribution, instead of applying neural networks as standalone coupling algorithms, we use them to enhance existing generic coupling algorithms. Through online learning, we enable algorithms to adapt to the coupling signals of RVPs, with the goal of further reducing coupling errors. As with the concept of hybrid models [27], by starting from generic coupling algorithms, this approach aims to leverage as much a priori knowledge as possible before utilizing machine learning methods for the further improvement of our coupling algorithms. This is a key difference to methods that require mandatory online training, such as in [28], where an FIR filter serves as the foundation for the learning approach or the coupling algorithm proposed by the authors of [29], which employs a PD controller function as the basis for training. Following that, in Section 2.4, we extend the frequency domain analysis of the entire coupling process of RVPs as well as the method to determine optimal coupling algorithms based on it, which both originate from our previous publication [26] to conform to our defined general form of coupling algorithms.

A well-known issue with coupling algorithms is the limitation in their usability for coupling signals with changing frequency components, such as those caused by discontinuities, especially stateful methods, such as the PD controller mentioned above, as well as other approaches, such as the model-based coupling element utilized in [30] and analyzed in [31] or the sliding-mode observer for latency compensation proposed by the authors of [32,33]; these perform poorly when applied to such signals, as they tend to impose oscillations.

For our stateless coupling algorithms, we address this issue in Section 2.5 by introducing a novel method for detecting and handling discontinuities in coupling signals. This approach aims to prevent significant coupling errors that would occur when applying a coupling algorithm at a discontinuity and, thus, increase the range of applicability of these coupling algorithms.

After the implementation of the entire coupling method using the FMI standard in Section 2.6, we demonstrate our proposed coupling method in Section 3 by applying it to an RVP of a hybrid electric vehicle, which is used productively in industry, and by outlining the benefits achieved through the use of the coupling method. We conclude the contribution by summarizing the main results and identifying potential starting points for future work in Section 4.

2. Materials and Methods

2.1. Real-Time Co-Simulation for Real-Virtual Prototypes

Real-time co-simulation combines methods for coupling prototypes under existing real-time requirements, where the prototypes are spatially distributed and communicate over a network during run-time. Consequently, the data exchange of the prototypes of an RVP can be formulated as a real-time co-simulation [34].

The data exchange between prototypes under real-time conditions over a network, as depicted in Figure 1, occurs similarly to the data exchange between the simulation models in a co-simulation (see Section 1) at specific time points, represented by the sequence

$$t_n = n\Delta T \tag{1}$$

with the time t, constant macro step size, ΔT, and $n \in \mathbb{N}$. In addition to sampling, network effects, such as time-varying communication latency $\tau(t)$ and packet loss, affect the coupling signals negatively [35]. As a result, the ideal coupling equation

$$u(t) = y(t) \tag{2}$$

for an input, u, that is connected to an output, y, of another prototype is never fulfilled. Instead, due to the communication latency, it holds

$$u(t) = \hat{y}(t) = y_{n-k} \text{ with } t_n \leq t < t_{n+1} \tag{3}$$

and

$$k = \min\{s \in \mathbb{N} | s \geq \frac{\tau(t_n)}{\Delta T}\} \text{ with } \tau > 0 \tag{4}$$

Thereby, k stands for the number of macro steps delay, in the current macro step due to latency. For the RVPs in the literature, the dependency between ΔT and k from Equation (4) is resolved by choosing the macro time step in such a way that, on average, it holds that $k \leq 10$ [28,34–36]. Furthermore, as in offline co-simulation, the macro step size in a real-time co-simulation must be chosen to be much smaller than the Nyquist frequency to avoid aliasing [37].

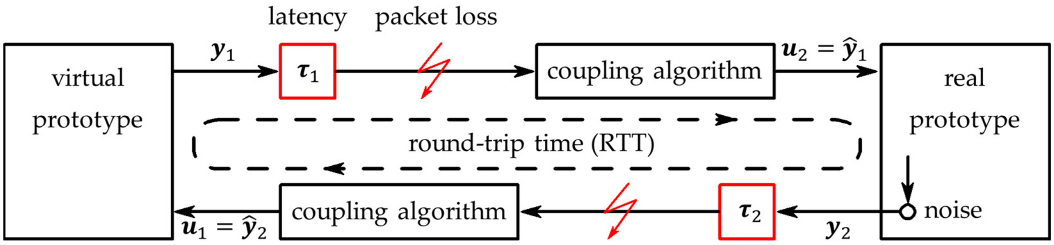


Figure 1. Schematic representation of a real-virtual prototype (RVP), consisting of one real and one virtual prototype, with a focus on the coupling faults during data exchange due to network effects, marked in red. Packet loss is indicated by the red flash. Based on [28].

The packet loss mentioned, which is inevitable during data exchange over an insecure network, such as the internet, can be interpreted as an additional short-term increase in latency. This is due to the fact that in the case of a single packet loss at t_n, the message received at t_{n-1} remains the latest until a new message arrives at t_{n+1}, increasing the latency by ΔT. Real-time violations of the virtual prototypes can also lead to temporary latency increases. This occurs when a virtual prototype fails to complete its calculations before the next macro step, causing the message to be sent late and, subsequently, received later by a coupled prototype. The authors of [38] discuss these time discrepancies between real and virtual prototypes in detail and propose a mechanism for detecting them. Additionally, noise occurring during signal measurement in real prototypes affects the roundtrip time between prototypes, as filters, which are optimized for the sensors used, are often utilized to compensate for the noise. These filters also contribute to the latency, as they introduce a phase shift to the coupling signal. Consequently, the network errors can be represented in a simplified form as pure variable latency.

The discrepancy between the ideal coupling described in Equation (2) and the actual coupling with latency represented by Equation (3) is referred to as the coupling error. For an arbitrary coupling signal, it is defined as

$$e(t) = d(y(t), \hat{y}(t)) \text{ with } \; t_n \leq t < t_{n+1} \tag{5}$$

in this work and it is calculated using a suitable distance function, $d(\cdot)$. Many different distance functions are possible. For instance, the authors of [28] chose the absolute error for it.

In order to reduce the coupling error and, thus, improve the significance of the results achieved with RVPs, coupling algorithms are utilized. These algorithms aim to enhance the accuracy of the coupling process by compensating for the latency and by reconstructing the continuous coupling signal. In this work, stateless, linear coupling algorithms of the general form

$$\hat{y}(t) = \sum_{i=0}^{m-1} a_i y_{n-k-i} + \sum_{l=0}^{m-1} A_l y_{n-k-l} \frac{t - t_n}{\Delta T} \text{ with } t_n \leq t < t_{n+1} \tag{6}$$

are considered. Both terms are an autoregressive model [39]. The first compensates for the latency by estimating the signal value for the current macro time t_n, and the second reconstructs the continuous signal up to the beginning of the next macro time point, which is exactly the task of an extrapolation algorithm used in non-iterative offline co-simulation. The reconstruction is necessary to allow for calculating input values for the prototypes between the time points defined by the macro step size, as the internal micro step size of virtual prototypes or the frequency of actuation of real prototypes may differ from the macro step size chosen for the RVP. Coupling algorithms that correspond to the form in Equation (6) differ in the values of their weighting vectors, $\boldsymbol{a}$ and $\boldsymbol{A}$, as well as in the number of past signal values used, m.

As for example in [31], the commonly used naïve coupling algorithm zero-order-hold (ZOH) is defined as

$$\hat{y}(t) = y_{n-k} \text{ for } t_n \leq t < t_{n+1} \tag{7}$$

and it estimates the current signal course, $\hat{y}(t)$, by simply taking the latest available signal value (see Equation (3)), making it easy to implement; therefore, it is commonly used. It is derivable from Equation (6) using the parameters $a = 1$, $A = 0$, and $m = 1$. The first-order-hold (FOH) coupling algorithm that is sometimes used in co-simulation for physical signals that do not contain discontinuities [37] is also extendable for compensating latencies in real-time simulation. For FOH, the estimated signal course is given by

$$\hat{y}(t) = y_{n-k} + (y_{n-k} - y_{n-k-1}) \left(k + \frac{t - t_n}{\Delta T} \right) \text{ for } t_n \leq t < t_{n+1}, \tag{8}$$

which conforms to Equation (6) for the parameters $\boldsymbol{a} = [k+1, -k]^T$, $\boldsymbol{A} = [1, -1]^T$, and $m = 2$. The signal is continued based on the first derivative calculated with the backward differentiation coefficient. Besides ZOH and FOH, the stateless coupling algorithms from the literature, such as the one from [28], that were introduced in Section 1 also correspond to the general form in Equation (6). However, stateful algorithms [29,30,32], often based on an autoregressive model, do not conform to this form; therefore, the methods presented in this contribution are not applicable to them.

2.2. *Error Space Extrapolation*

In this contribution, we first propose a novel method for constructing a new class of coupling algorithms that, like ZOH and FOH, are generically applicable to the coupling signals of RVPs, as they do not require system knowledge of the prototypes of the RVP. Instead of compensating for latency by estimating signal characteristics, such as the first derivative, the proposed method explicitly focuses on the primary objective of a coupling algorithm (see Section 2.1), which is minimizing the coupling error.

Starting from a set of curves, $f_p(y_{k,m}, k, t)$, the method chooses the curve $f_{\hat{p}}(y_{k,m}, k, t)$ that shall be used for extrapolation until the next macro step. When applying the selected curve, the extrapolated coupling signal results in

$$\hat{y}(t) = f_{\hat{p}}\left(\boldsymbol{y}_{k,m}, k, t\right) \text{with } \boldsymbol{y}_{k,m} = \begin{bmatrix} y_{n-k} \\ y_{n-k-1} \\ \vdots \\ y_{n-k-m+1} \end{bmatrix} \text{for } t_n \leq t < t_{n+1}, \tag{9}$$

whereby the parameter vector $\hat{p} \in \mathcal{D}_p$ defines the shape of the curve. The objective of the method is to select $f_{\hat{p}}(t)$ in such a manner that the estimated coupling error from Equation (5) at the end of the current macro time step

$$\hat{e}_{n+1} = d(y_{n+1}, \hat{y}_{n+1}) \tag{10}$$

is minimized based on an appropriate distance function, $d(\cdot)$.

Given a set of available signal curves, f_p, with $p \in \mathcal{D}_p$ and the distance function, the method predicts (at each macro time step) the value $\hat{p}$ of the parameter p that minimizes the absolute value of the estimated coupling error, $\hat{e}_{n+1}$, resulting in

$$\hat{p} : \min_{p \in \mathcal{D}_p} \hat{e}_{n+1}(p) = \min_{p \in \mathcal{D}_p} d\left(y_{n+1}, f_p\left(\boldsymbol{y}_{k,m}, k, t_{n+1}\right)\right). \tag{11}$$

The core of the method is the estimation of $\hat{p}$ through the analysis and extrapolation of the past signal curve not in the signal space but in the error space. The procedure is divided into a sequence of four steps that are executed once at every macro time step.

1. **Estimation of (hypothetical) coupling errors from previous macro steps.** Based on the given historical time series of the coupling signal, $\boldsymbol{y}_{k,m}$, at each of the last $q \geq 1$ time points, a set of possible extrapolation errors $e_i(p)$ with $n - k - q < i \leq n - k$ is determined. This can be carried out either analytically according to Equation (10) or by approximation, where $e_i(p)$ is evaluated exactly for s_i supporting points $p_{i,j}$ and the curve of $e_i(p)$ between the supporting points are determined by curve fitting to functions g_i.
2. **Estimate future coupling errors.** Using the computed q error curves, $e_i(p)$ and an appropriate extrapolation method, E, the error curve $\hat{e}_{n+1}(p)$ for the next time step is estimated by applying E to $e_i(p)$. Similar to the previous step, the error curve $\hat{e}_{n+1}(p)$ can either be calculated analytically or approximated by evaluating $\hat{e}_{n+1}(p)$ for s_{n+1} supporting points $p_{n+1,j}$ and determining the curve of $\hat{e}_{n+1}(p)$ between the supporting points $e_{n+1}\left(p_{n,j}\right)$ through curve fitting using the functions g_{n+1}.
3. **Find the smallest future coupling error.** With the estimation of the future error curve, $\hat{e}_{n+1}(p)$, $\hat{p}$ is determined by finding an approximation of the solution of Equation (11).
4. **Extrapolate the coupling signal**. By using $\hat{p}$, the future coupling signal curve is approximated according to Equation (9).

This general method can be varied in the following aspects:

- The choice of the set of curves, $f_p(t)$;
- The choice of the distance function, $d(\cdot)$;
- The number, q, of past time points considered;
- The choice of the method, E, for estimating $\hat{e}_{n+1}(p)$ using $e_i(p)$;
- The choice of the method for solving Equation (11).

If the error curves $e_i(p)$ or $\hat{e}_{n+1}(p)$ cannot be computed analytically, two additional aspects need to be determined:

- The number, s, of supporting points at which the error curves will be precisely evaluated;

- The choice of functions, g_i, for curve fitting between the supporting points.

Depending on how these variation points are chosen, many of the steps in the proposed method can be combined, resulting in a coupling algorithm, the execution of which requires little computing time. This will be demonstrated below by deriving a specific coupling algorithm using the proposed method.

For our coupling algorithm, a chosen first-degree polynomial defines the set of curves as

$$f_p\left(\boldsymbol{y}_{k,m}, k, t\right) = p_0 + p_1\left(k + \frac{t - t_n}{\Delta T}\right) \text{ for } t_n \leq t < t_{n+1}. \tag{12}$$

Thus, with Equation (9), the extrapolated coupling signal curve results in

$$\hat{y}(t) = y_{n-k} + \hat{p}_1\left(k + \frac{t - t_n}{\Delta T}\right) \text{ für } t_n \leq t < t_{n+1}. \tag{13}$$

When choosing the last known signal value as a point on the polynomial $\hat{p}_{n,0} = y_{n-k}$, make sure the estimated signal curve is continuous. In order to determine $\hat{p}_1$ using extrapolation in the error space, the Taylor series expansion

$$\hat{e}_{n+1}(p) = e_{n-k}(p) + (k+1)\Delta T \frac{d}{dt} e_{n-k}(p) + \frac{1}{2}((k+1)\Delta T)^2 \left(\frac{d}{dt}\right)^2 e_{n-k}(p) + \ldots \tag{14}$$

is chosen as method E, using the backward difference quotient to approximate the derivatives. For the specific coupling algorithm derived here, the Taylor series expansion is stopped after the first order, whereby

$$\hat{e}_{n+1}(p) = e_{n-k}(p) + (k+1)(e_{n-k}(p) - e_{n-k-1}(p)) \tag{15}$$

applies. By choosing the absolute error as the distance function, $d(\cdot)$, the required error curves can be obtained analytically as

$$e_i(p) = \hat{y}_i(p) - y_i = y_{i-k-1} + p_1(k+1) - y_i, \tag{16}$$

eliminating the need for curve fitting in this case. Thus, the $q = 2$ past error curves required by Equation (15) result in

$$\begin{aligned} e_{n-k}(p) &= y_{n-2k-1} + p_1(k+1) - y_{n-k} \text{ and} \\ e_{n-k-1}(p) & \\ &= y_{n-2k-2} + p_1(k+1) - y_{n-k-1}. \end{aligned} \tag{17}$$

This allows for the analytical computation of the absolute minimum of the future coupling error by solving Equation (11) with $\hat{e}_{n+1}(p) \overset{!}{=} 0$. With the solution

$$\hat{p}_1 = \frac{k+2}{k+1} y_{n-k} - y_{n-k-1} - \frac{k+2}{k+1} y_{n-2k-1} + y_{n-2k-2} \tag{18}$$

and Equation (13), the derived coupling algorithm is defined as

$$\begin{gathered} \hat{y}(t) = y_{n-k} + \left(\tfrac{k+2}{k+1} y_{n-k} - y_{n-k-1} - \tfrac{k+2}{k+1} y_{n-2k-1} + y_{n-2k-2}\right)\left(k + \tfrac{t-t_n}{\Delta T}\right) \\ \text{for } t_n \leq t < t_{n+1}. \end{gathered} \tag{19}$$

In this contribution, we name this coupling algorithm error space extrapolation (EROS). Just like ZOH and FOH, EROS is a generic coupling algorithm that conforms to the general form of the coupling algorithms of Equation (6) using the parameters

$$\boldsymbol{a} = \left[1 + \frac{k+2}{k+1}, -1, \ldots, -\frac{k+2}{k+1}, 1\right]^T, \; \boldsymbol{A} = \left[\frac{k+2}{k+1}, -1, \ldots, -\frac{k+2}{k+1}, 1\right]^T \text{ and } m = k+3. \tag{20}$$

It does not require any parametrization and can, therefore, be applied without system knowledge about the RVP or its coupling signals. Furthermore, the calculation formula of EROS is directly dependent on k, making it suitable for variable latency while keeping the computational effort constant, as it does not need to be applied recursively. Another interesting property of EROS is that the indices of the m past signal values used increase with increasing k. This fulfills a common rule of thumb in forecasting to "Look Back Twice as Far as You Look Forward" [40]. In Section 2.4, EROS is analyzed in more detail.

Certainly, countless other coupling algorithms with different properties (see Section 2.4) can be constructed using the presented method. However, in this contribution, our focus is specifically on EROS.

2.3. Adding Learning Ability to Coupling Algorithms by utilizing Feedforward Neural Networks

In a previous contribution [26], we introduced the concept of utilizing feedforward neural networks (FFNNs) as a coupling algorithm that can adapt to the coupling signal of RVPs through online learning. Thus, the FFNN coupling algorithm can predict the behavior of coupling signals resulting from the nonlinear behavior of the coupled prototypes, such as discontinuities, which is not achievable with linear coupling algorithms. In this contribution, we revisit this approach. However, we extend and generalize the method by constructing the FFNN in such a way that its initial input-output behavior can be chosen to be exactly equal to the behavior of any algorithm conforming to the general rule from Equation (6). In this way, we enhance coupling algorithms, such as the newly proposed EROS algorithm from Section 2.2, by online learning capabilities.

The starting point is a standard FFNN consisting of multiple fully connected layers. As defined in [41], the h-th layer computes its output vector

$$\boldsymbol{x}^h = \sigma\left(\mathbf{W}^h \boldsymbol{x}^{h-1} + \boldsymbol{\Theta}^h\right) \tag{21}$$

using the input vector $\boldsymbol{x}^{h-1}$ with a generally nonlinear activation function, $\sigma(\cdot)$ per neuron. The dimension of the weight matrix, $\mathbf{W}^h$, and the offsets vector, $\boldsymbol{\Theta}^h$, depends on the dimension of the input vector and the number of neurons, which is equal to the dimension of the output vector of the h-th layer. The input vector of the FFNN is the m most recent past signal values of the coupling signal, and the output is the predicted signal value for the current macro time step, $\hat{y}_n$.

As early as 1996, the authors of [42] observed that any arbitrarily large FFNN of the form in Equation (21) that is used for predicting univariate time series could be simplified as the autoregressive model

$$\hat{y}_n = c + \sum_{i=1}^{m} a_i y_{n-i} \tag{22}$$

when the linear activation function, $\sigma_{lin}(x) = x$, is used in each neuron of every layer. In this case, the vector $\boldsymbol{a}$ and the constant c correspond exactly to the weights and offset of the individual neuron. Since Equation (22) is an autoregressive model, it is the same as the first term of the generic form in Equation (6). Consequently, the latency compensation term of any generic coupling algorithm conforming to Equation (6) is transformable into an FFNN consisting of a single neuron and linear activation function. Thereby, the offset vector, $\boldsymbol{\Theta}^h$, of the single neuron will be set to zero as it is commonly assumed that $c = 0$ for all coupling algorithms, as otherwise, a coupling error would occur for the simple case of applying the coupling algorithm to a constant signal.

In order to increase the complexity of the FFNN and, thus, enhance the FFNN's ability to adapt to coupling signals arising from the complex nonlinear behavior of an RVP, the method of "Network Morphism" [43] is used. This involves increasing the number of neurons and layers, as well as introducing nonlinear activation functions within the FFNN while preserving its input-output behavior. First, the layers and neurons with linear activation functions are added by utilizing the net2net method [44]. Then, the nonlinear behavior is introduced into the network by replacing single neurons with a linear activation function with two neurons with the LRELU activation function

$$\sigma_{lrelu}(x) = \begin{cases} x \text{ if } x \geq 0 \\ \alpha x \text{ if } x < 0 \end{cases} \tag{23}$$

with the commonly used factor $\alpha = 0.01$ [45] via the relation

$$\sigma_{lin}(x) = \frac{1}{1+\alpha}(\sigma_{lrelu}(x) + \sigma_{lrelu}(-x)). \tag{24}$$

In order to provide the FFNN with enough flexibility to adapt to individual coupling signals while keeping computational complexity and online training time low, the number of neurons in the final FFNN architecture must be chosen carefully. In this contribution, the first layer consists of four, each fully connected to the input vector composed of the past signal values of the coupling signal. The second layer comes with two neurons before the scalar prediction of the time series is computed as the output of the output layer with a single neuron. All the neurons of the first and the last layers use the linear activation function.

If the coupling signal that the FFNN coupling algorithm is applied to contains discontinuities, the LRELU activation function should be selected for the neurons of the second layer. This choice enables the FFNN algorithm to have nonlinear behavior by permitting it to learn to switch between different behaviors, thereby predicting discontinuities. However, in the absence of discontinuities, the linear activation function can also be utilized in the second layer, resulting in linear behavior for the FFNN, thus simplifying the analysis of its behavior (see Section 2.4).

The FFNN coupling algorithm only addresses the compensation for latency, which is the first part of Equation (6), and not the reconstruction of the continuous signal; for that, two options are available. Either the reconstruction part of the coupling algorithm that was the basis for the FFNN is used or ZOH is simply taken to prevent the amplification of potential discontinuities (see Section 2.4).

The resulting FFNN is already applicable as a coupling algorithm without training, as its behavior exactly corresponds to the generic coupling algorithm (e.g., ZOH, FOH, or EROS), which is used as the initial parametrization of the network. In addition, based on this already good initial parameterization, it is possible to train the FFNN online during the run-time of an RVP with little computational effort to adapt to a specific coupling signal using its past course as training data and by utilizing highly efficient training procedures specifically designed for the structure of FFNNs, e.g., as discussed in [46]. In [26], we demonstrated that this approach enables the prediction of discontinuities in the coupling signal that results from the nonlinear behavior of the coupled prototypes.

2.4. Coupling Process Analysis and Optimization in the Frequency Domain

As explained in Section 2.1, the system behavior of RVPs is influenced by coupling faults, such as latency, and the coupling algorithms used to minimize the coupling error. In the following, the influence of the overall coupling process consisting of coupling faults and a coupling algorithm on the system behavior of RVPs is examined in the frequency domain. This analysis aims to derive an upper bound on the bandwidth of the coupling signal for which the generic coupling algorithms are applicable. In addition, it is the basis with which to develop a method for calculating optimal coupling algorithms with respect to the bandwidth of the coupling signals of a specific RVP.

While the authors of [37], along with ourselves in our own publication [26], conduct an analysis of the coupling process, including coupling algorithms, the former focuses specifically on non-iterative offline co-simulation. Consequently, it does not consider the coupling faults caused by the real-time requirement of RVPs, which are crucial for the analysis of RVPs, as they represent the primary negative effect on their behavior (see Section 2.1). The latter only considers coupling algorithms of the form in Equation (22) and not the general form of linear algorithms, including the reconstructions in Equation (6), which limits the analysis and can lead to imprecise results. Therefore, the analysis and the calculation of the optimal coupling algorithms are based on it, which both originate from [23]; these are generalized and are hereby improved upon in the following section to provide a more precise estimation of the RVP's behavior due to coupling faults and coupling algorithms.

2.4.1. Transforming the Coupling Process into the Frequency Domain

When assuming the inputs and outputs of all the prototypes of an RVP are time-continuous, it is reasonable to interpret the coupling process as a time-continuous element. Consequently, the Laplace transform is applicable for calculating its transfer function. As shown in Figure 2, the transfer function of the coupling process

$$G_p(s) = G_r(s)G_l(s)G_s(s) \tag{25}$$

is split into three parts: sampling, latency, and the coupling algorithm, including compensation for the latency and reconstruction of the continuous signal. Each part is calculated independently.

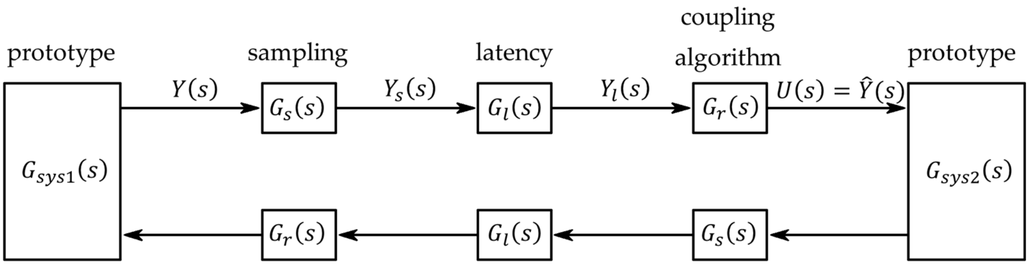


Figure 2. Block diagram of an RVP in the frequency domain. The transfer behavior of the two coupled prototypes is represented by $G_{sys1}(s)$ and $G_{sys2}(s)$. Additionally, the transfer behavior of the network effects in each direction (see Section 2.1) is described by $G_s(s)$ for the sampling, $G_l(s)$ for the latency, and $G_r(s)$ for the coupling algorithm. The signals between the blocks have been assigned specific names, which are utilized in the derivation of the overall transfer function.

The effects of sampling on a continuous signal in the frequency domain are derived in detail in [47]. There, the signal sampled with the macro step size, ΔT, is represented as

$$y_s(t) = \sum_{n=-\infty}^{\infty} y_n \delta(t - n\Delta T) \tag{26}$$

using the Dirac function, $\delta(\cdot)$. The transfer function of this sampling results in

$$G_s(s) = \frac{Y_s(s)}{Y(s)} = \frac{1}{\Delta T} \tag{27}$$

under the assumption that the macro step that was chosen is small enough to avoid aliasing, which is not a limitation, as aliasing must be avoided anyway to allow for the correct operation of an RVP (see Section 2.1). The Laplace transformation of the sampled signal can also be expressed in terms of the sampled signal values $y_n = y(n\Delta T)$ as

$$Y_s(s) = \sum_{n=0}^{\infty} y_n e^{-sn\Delta T} \tag{28}$$

by applying the Laplace transform pair $\delta(t - n\Delta T) \circ\!\!-\!\!\bullet\, e^{-sn\Delta T}$.

For the second part of the coupling process transfer function—the transfer function of the latency—it holds that

$$G_l(s) = \frac{Y_l(s)}{Y_s(s)} = e^{-sk\Delta T} \tag{29}$$

as a constant latency $k\Delta T$ (see Section 2.1) has the effect of a time shift in the frequency domain.

In order to calculate the Laplace transformation of the coupling algorithms, which is the third part of the coupling process, the general form of the coupling algorithms from Equation (6)

$$\hat{y}(t) = \underbrace{\sum_{i=0}^{m-1} a_i y_{n-k-i}}_{\hat{y}_1(t)} + \underbrace{\sum_{l=0}^{m-1} A_l y_{n-k-l} \frac{t-t_n}{\Delta T}}_{\hat{y}_2(t)} \text{ with } t_n \leq t < t_{n+1}, \tag{30}$$

is utilized. It generalizes, among others, the commonly used methods ZOH and FOH and the newly introduced EROS coupling algorithm from Section 2.2, as well as the linear FFNN from Section 2.3. Due to the piece-wise definition of $\hat{y}(t)$, applying the Laplace transform, together with $y(t) = 0$ for $t < 0$, leads to

$$\hat{Y}(s) = \sum_{n=0}^{\infty} \int\limits_{n\Delta T}^{(n+1)\Delta T} \hat{y}(t) e^{-st} dt. \tag{31}$$

The linearity property of the Laplace transform allows us to solve this by independently transforming the two terms $\hat{y}_1(t)$ and $\hat{y}_2(t)$ from Equation (30). For the first term, after performing the integration, this results in

$$\mathcal{L}\{\hat{y}_1(t)\} = \sum_{i=0}^{m-1} a_i e^{-(k+i)s\Delta T} \sum_{n=0}^{\infty} y_n \left[-\frac{1}{s} e^{-st}\right]_{n\Delta T}^{(n+1)\Delta T} = \sum_{l=0}^{m-1} a_i e^{-is\Delta T} \frac{1-e^{-s\Delta T}}{s} \underbrace{e^{-ks\Delta T}}_{G_l(s)} \underbrace{\sum_{n=0}^{\infty} y_n e^{-sn\Delta T}}_{Y_s(s)} \tag{32}$$

and for the second term, it leads to

$$\begin{aligned} \mathcal{L}\{\hat{y}_2(t)\} &= \sum_{l=0}^{m-1} A_l e^{-(k+l)s\Delta T} \sum_{n=0}^{\infty} y_n \left[-\tfrac{1}{s^2\Delta T} e^{-st} - \tfrac{1}{s\Delta T} t e^{-st} + \tfrac{1}{s} n e^{-st}\right]_{n\Delta T}^{(n+1)\Delta T} \\ &= \sum_{j=0}^{m-1} A_l e^{-ls\Delta T} \tfrac{1-(1+s\Delta T)e^{-s\Delta T}}{s^2\Delta T} \underbrace{e^{-ks\Delta T}}_{G_l(s)} \underbrace{\sum_{n=0}^{\infty} y_n e^{-sn\Delta T}}_{Y_s(s)}. \end{aligned} \tag{33}$$

After adding Equations (32) and (33), the transfer function of the general form of the coupling algorithms is calculated as

$$G_r(s) = \frac{\hat{Y}(s)}{Y_l(s)} = \sum_{i=0}^{m-1} a_i e^{-is\Delta T} \frac{1-e^{-s\Delta T}}{s} + \sum_{l=0}^{m-1} A_l e^{-ls\Delta T} \frac{1-(1+s\Delta T)e^{-s\Delta T}}{s^2\Delta T} \tag{34}$$

using $Y_l(s) = G_l(s)Y_s(s)$ (refer to Figure 2).

Combining the three parts of the transfer function of the coupling process of RVPs, taking into account coupling faults and the coupling algorithm, results in

$$G_p(s) = G_r(s)G_l(s)G_a(s) = \left(\sum_{l=0}^{m-1} a_l e^{-ls\Delta T} \frac{1-e^{-s\Delta T}}{s\Delta T} + \sum_{j=0}^{m-1} A_j e^{-js\Delta T} \frac{1-(1+s\Delta T)e^{-s\Delta T}}{s^2\Delta T^2}\right) e^{-ks\Delta T} \tag{35}$$

By substituting the parameter vectors $\boldsymbol{a}$ and $\boldsymbol{A}$ from Equations (7), (8) and (19) into Equation (35), the transfer function for the coupling process is calculated for the coupling algorithms ZOH, FOH, and EROS. The resulting transfer functions are depicted as Bode plots in Figure 3 for an exemplary latency of $k = 3$ macro time steps. An ideal coupling algorithm would compensate for the phase error caused by the latency, resulting in a neutral magnitude $|G_P(j\omega_m)| = 1$ and a zero-phase $\left|\angle G_p\left(j\omega_p\right)\right| = 0^{\circ}$ in response to the coupling process for all frequencies below the Nyquist frequency. FOH and EROS achieve this at low frequencies, but this comes at the cost of high-magnitude amplification for higher frequencies due to the waterbed effect [48]. This is the reason why these methods are not applicable when high frequencies (for example, due to discontinuities) are present in the coupling signal. ZOH cannot compensate for the latency at all but it also does not amplify high frequencies.

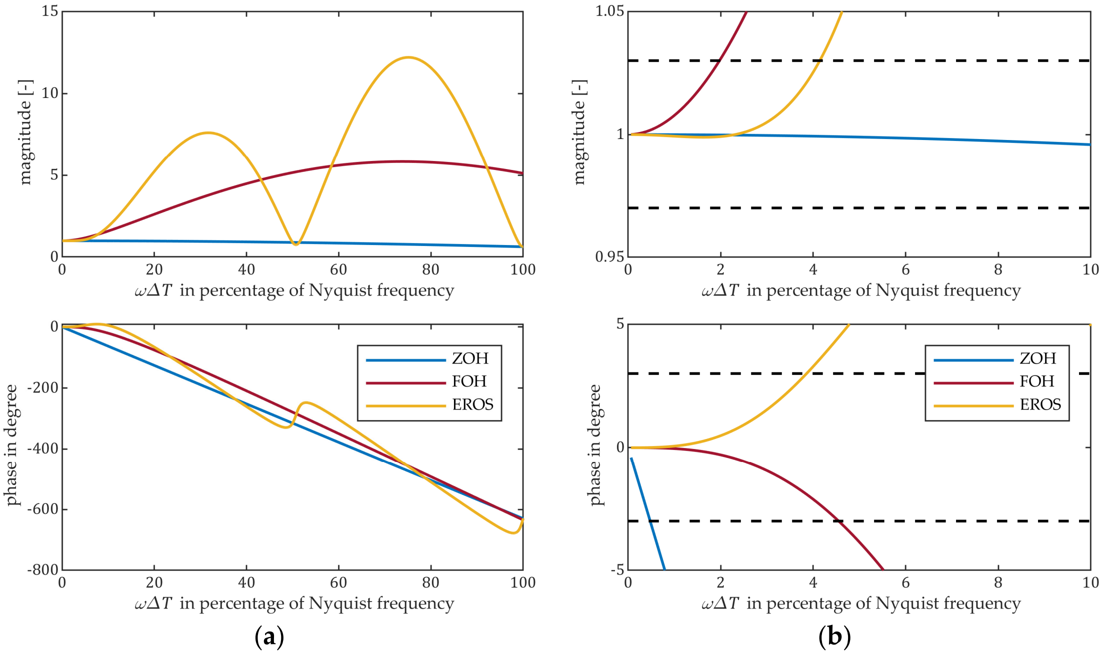


Figure 3. Bode plot of the coupling process of RVPs for an exemplary latency of $k = 3$, normalized to the frequency with respect to the macro step as a percentage of the Nyquist frequency (compare to [37] for non-iterative co-simulation) for the coupling algorithms zero-order-hold (ZOH), first-order-hold (FOH), and error space extrapolation (EROS): (**a**) overview; (**b**) focus on low frequencies. Magnitude and phase limits as introduced by the authors of [37] are shown as a dashed line.

In order to define the frequency ranges for which the coupling algorithms are suitable and to compare different coupling algorithms, the authors of [37] introduce heuristic limits for the maximum deviation from the ideal transfer behavior. For the magnitude gain, they choose $|G_P(j\omega_m)| < 1.03$, and for the phase shift, $\left|\angle G_p\left(j\omega_p\right)\right| < 3^{\circ}$, resulting in an upper bound of the frequency range $\overline{\omega\Delta T} = \min\{\overline{\omega_m\Delta T}, \overline{\omega_p\Delta T}\}$, for which a coupling algorithm is to be used. Table 1 shows this upper bound of the frequency range for the considered coupling algorithms for different latency values. For lower latency values, the frequency range for all coupling algorithms expands, whereas for higher latency values, it significantly narrows. The newly proposed EROS algorithm is applicable in the widest frequency range, primarily due to its good balance between magnitude and phase errors.

Table 1. Upper bounds of the magnitude and phase conditions for different coupling algorithms as a percentage of the Nyquist frequency for various latency times, $k\Delta T$. The resulting upper bound of the frequency range $\overline{\omega\Delta T} = \min(\omega_a\Delta T,\ \overline{\omega_p\Delta T})$ is marked in bold in each case.

	Latency *k*=0		**Latency *k*=1**		**Latency *k*=3**		**Latency *k*=6**	
	$\overline{\omega_m\Delta T}$	$\overline{\omega_p\Delta T}$	$\overline{\omega_m\Delta T}$	$\overline{\omega_p\Delta T}$	$\overline{\omega_m\Delta T}$	$\overline{\omega_p\Delta T}$	$\overline{\omega_m\Delta T}$	$\overline{\omega_p\Delta T}$
ZOH	27.13	**3.33**	27.13	**1.11**	27.13	**0.48**	27.13	**0.25**
FOH	**8.66**	18.20	**4.01**	9.10	**1.97**	4.55	**1.11**	2.57
EROS	18.15	**16.28**	8.75	**7.61**	4.14	**3.86**	2.29	**2.26**

The analysis shown in this section is limited to coupling signals with a constant latency, whereas, in general, the latency in RVPs is time-varying due to jitter and packet losses (see Section 2.1). However, when setting up an RVP, efforts are made to keep the latency as close to being constant as possible, as a constant latency increases the comparability of the results from multiple runs of the RVP. Moreover, as we previously presented in [26], the transfer function of the coupling process also allows for a stability analysis of RVPs using the Nyquist stability criterion, provided that a transfer function of each coupled prototype is available and the latency is constant. An approach to achieving a constant latency is by carefully selecting the macro step size (see Equation (4)) or optimizing the alignment of the sampling sequences between different prototypes, as demonstrated in our patented method [49].

2.4.2. Calculating an Optimal Coupling Algorithm

In contrast to the previous section, where the analysis method was used to derive the validity ranges for existing coupling algorithms, it is now reversed and utilized to determine an optimal coupling algorithm for a given bandwidth of a specific coupling signal of an RVP.

The idea is to formulate an optimization problem containing the derived transfer function of the coupling process from which the vectors $\boldsymbol{a}$ and $\mathbf{A}$ of the general form of a coupling algorithm (Equation (6)) are obtained. The optimization problem reads as

$$\min_{\boldsymbol{a},\boldsymbol{A}} J(\boldsymbol{a},\boldsymbol{A}) = \alpha J_m + \beta J_p + \gamma J_r, \text{ subject to } G_p(0) = 1, \tag{36}$$

whereby the constraint $G_p(0) = 1$ with $s = j\omega = 0$ ensures that a constant coupling signal is accurately reproduced by the resulting coupling algorithm. The cost function $J(\boldsymbol{a},\boldsymbol{A})$ consists of the three summands:

$$\begin{aligned} J_m &= \int_{\omega_l}^{\omega_u} \frac{1-|G_p(j\omega)|}{\omega_u-\omega_l}\mathrm{d}\omega,\ J_p = \int_{\omega_l}^{\omega_u} \frac{\angle G_p(j\omega)}{\omega_u-\omega_l}\mathrm{d}\omega \text{ and} \\ J_r &= \int_0^{\omega_l} \max(|G_p(j\omega)|,1)\mathrm{d}\omega - \omega_l + \int_{\omega_u}^{\frac{2\pi}{\Delta T}} \max\left(|G_p(j\omega)| - \left(\frac{\omega}{\omega_u}\right)^{\frac{1}{2}r},0\right)\mathrm{d}\omega \end{aligned} \tag{37}$$

each of which penalizes a different aspect of the transfer function of the overall coupling process to ensure that the coupling faults are compensated for by the bandwidth of the coupling signal $\omega_b \in [\omega_l,\ \omega_u]$.

- J_m penalizes a derivation from the neutral magnitude $|G_p(j\omega)| = 1$ within the bandwidth of the coupling signal;
- J_p penalizes a derivation from zero-phase shift $\angle G_p(j\omega) = 0°$ within the bandwidth;
- J_r penalizes a magnitude amplification outside of the bandwidth of the coupling signal. For frequencies lower than the bandwidth, it simply penalizes a magnitude that is greater than one. For frequencies above the bandwidth of the coupling signal, J_r penalizes if the amplification due to the coupling algorithm is not compensated for by the natural damping of the RVP. In order to estimate the natural damping, no transfer function of the coupled system is required; instead, the relative degree, r, which can be estimated (see Section 3.2), is the only system information necessary.

The weighting parameters are chosen as $\alpha = 1$, $\beta = 0.01$, and $\gamma = 1000$. The large value of γ ensures that the third summand is prioritized and, thus, an amplification outside of the bandwidth of the coupling signal is prevented. Following the argumentation for the heuristic limits by [37], the ratio of $\alpha = 100\beta$ is set within the bandwidth of the signal such that a phase difference of 1° generates the same cost as a magnitude error of 1%.

2.5. Enhancing Linear Coupling Algorithms with Preceding Discontinuity Detection

In this section, we propose a novel method of utilizing preceding discontinuity detection before applying linear coupling algorithms. This approach aims to expand the range of applicability of such algorithms.

As analyzed in Section 2.4, linear coupling algorithms that can compensate for phase errors in the low-frequency range of the coupling signal, such as FOH and EROS, tend to amplify high frequencies. Therefore, these algorithms are not usable in RVPs when the coupling signal contains high frequencies. This limitation also applies when high frequencies occur only sporadically, for example, due to discontinuities in the signal, although the chosen coupling algorithm would be suitable for the major parts of the signal. In order to be able to use a coupling algorithm other than ZOH, despite occasional discontinuities or short sections where high frequencies may occur within the coupling signal, we propose the following to detect and respond to high frequencies in a signal online before coupling errors arise from a coupling algorithm such as FOH or EROS. In order to achieve this, the first step is to detect whether a sudden frequency change occurs between the last and current signal values at the edge of the time series. Subsequently, the coupling algorithm is temporarily switched to avoid significant coupling errors.

2.5.1. Online Frequency Spectrum Estimation

In order to detect a sudden frequency change in the newest signal value of a coupling signal, the frequency spectrum at the edge of the time series is estimated online during the run time of an RVP in each macro time step. For this, the discrete Fourier transform (DFT) [50] is applied to the last N values of a coupling signal, y, transforming it into its q frequency components:

$$Y_q = DFT(y) = \sum_{n=1}^{N} y_n \cdot e^{-\frac{j2\pi}{N}qn}. \tag{38}$$

Since the coupling signals in RVPs are real values in the time domain, they exhibit Hermitian symmetry in the frequency domain. This means that the result of the DFT of a real-valued signal, denoted as $rDFT(\cdot)$, only contains frequencies with positive frequency components. As a result, the number of frequency components, Y_q, is reduced to the first $\lfloor N/2 \rfloor + 1$ entries of the DFT, requiring the appropriate scaling of their magnitudes. It results in

$$Y = \frac{2}{N}|rDFT(y)|. \tag{39}$$

The length of the considered signal segment, N, influences the resolution of the frequency components in the DFT through the relationship $\omega_{res} = 2\pi/N$. A larger N increases the resolution but also increases the computation time of the DFT. Additionally, a larger N reduces the influence of the last signal value, which is being tested for a discontinuity regarding the individual frequency components. Therefore, we chose the relatively small value $N = 8$ for the discontinuity detection method presented here, which results in a DFT resolution of $\omega_{res} = \pi/4$, corresponding to 25% of the Nyquist frequency of the signal.

An important assumption made when applying the DFT to a signal segment is the periodic continuation of the segment. If this is not the case, high-frequency components are mistakenly added to the frequency spectrum due to the leakage effect [51]. In order to reduce this effect, the signal segment is multiplied by a window function before the transformation. Typically, window functions that make the respective edges of the signal segment approach zero are used, aiming to achieve the periodicity of the segment. However, for the developed method of detecting discontinuities in the coupling signals of RVPs, using such a window function is not beneficial, as it would heavily attenuate a detectable discontinuity at the end of a signal segment, making detection impossible. In order to counteract this effect, the considered signal segment is first shifted in the y-direction such that the last value of the segment is $y_n = 0$, and it is then multiplied by a half Hann window of the form

$$h(n) = 0.5\left(1 - \cos\left(\frac{\pi n}{N}\right)\right),\ 1 \leq n \leq N. \tag{40}$$

The half Hann window consists of the left half of the Hann window [52]. The parameter n represents the index of the signal segment, where a larger n corresponds to a more recent signal value.

The application of the half Hann window on a shifted signal segment containing a discontinuity at the current macro time point t_n is shown in Figure 4. The resulting signal segment (yellow) is periodically continuable without introducing an additional discontinuity between the periods, while the discontinuity in the last signal value that shall be detected is kept, as in the original signal segment (blue). For comparison, Figure 4 also shows the application of the regular Hann window on the nonshifted signal segment. The Hann window smooths out the discontinuity at the edge of the signal segment, which makes it no longer detectable in the frequency spectrum.

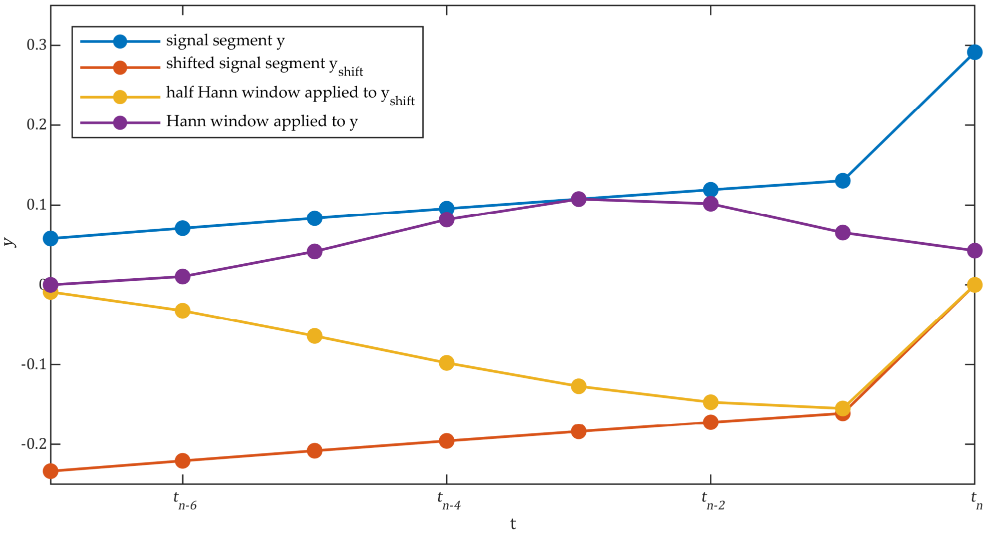


Figure 4. Demonstration of the application of the half Hann window on a shifted example signal segment with discontinuity, as compared to the application of the Hann window.

Applying a window function to a signal alters its magnitude and, consequently, affects its frequency spectrum. As proven by the authors of [51], this effect is compensated for by multiplying the frequency spectrum resulting from the DFT with the correction factor

$$A_w = \frac{N}{\sum_{n=1}^{N} h(n)}. \tag{41}$$

For the half Hann window from Equation (40), the correction factor results in $A_w = 1.7778$.

2.5.2. Criterion for Discontinuity

The method developed in this contribution detects a discontinuity in the current signal value of a coupling signal, y_n, if the condition

$$\sum_{k=2}^{5} Y_{q,n} > \xi \sum_{k=2}^{5} Y_{q,n-1} \tag{42}$$

is satisfied. The frequency spectrum, $Y_{q,n}$, of the signal at the current macro time step, t_n, is calculated using the real-valued DFT from Equation (39) on the shifted signal segment that was modified with the half Hann window (Equation (40)) and scaled with the correction factor (Equation (41)).

The frequency spectrum $Y_{q,n-1}$ refers to the corresponding spectrum calculated in the previous macro step t_{n-1}. The indices of the sum are chosen such that an increase in the magnitude of frequencies above 25% of the Nyquist frequency is considered within the criterion. The parameter ξ defines the sensitivity of the method and is set to $\xi = 5$ based on experimental analysis.

As an example, Figure 5 shows the frequency spectrum, $Y_{q,n}$, of the yellow signal segment from Figure 4 at the current macro time step, which contains a discontinuity, and the frequency spectrum, $Y_{q,n-1}$, of the same signal one macro time step earlier, which does not yet contain a discontinuity. As expected, the discontinuity increases the magnitude of all frequency components in the spectrum. In this case, the summed components above 25% of the Nyquist frequency are 9.3 times larger in the spectrum with the discontinuity, satisfying the criterion from Equation (42) and allowing for correctly detecting the discontinuity.

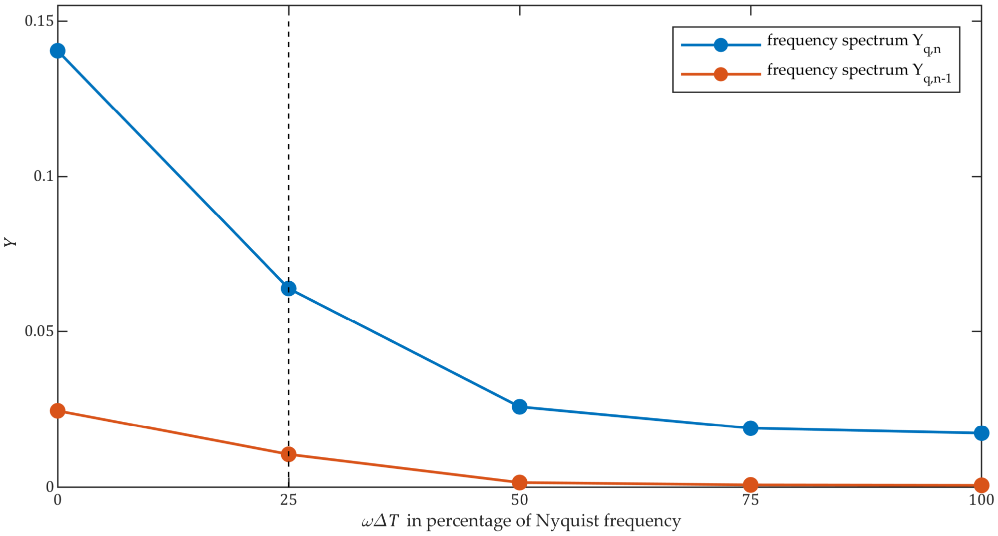


Figure 5. Resulting frequency spectrum for the yellow signal segment from Figure 4. $Y_{q,n}$ represents the frequency spectrum of the current macro step, where a discontinuity occurs, while $Y_{q,n}$ represents the spectrum from the previous macro step, where no discontinuity has occurred.

Once a discontinuity is detected, it needs to be addressed. In this contribution, we propose to switch the active coupling algorithm in the case of a discontinuity such that the signal values before and after the discontinuity are never simultaneously used for the prediction of any linear coupling algorithm. One way to achieve this is to switch to the naïve ZOH algorithm whenever a discontinuity has occurred within the last $m - 1$ time step, while m is the number of past signal values the current coupling algorithm uses. The idea of switching the coupling algorithm based on specific signal properties is also pursued in offline co-simulations to reduce reconstruction errors [53].

Figure 6 shows an example of how discontinuity detection with coupling algorithm switching works. The reference signal represents a sampled output of a real or virtual prototype of an RVP, with an exemplary latency of $k = 3$ macro time steps during the data transmission to another prototype. In order to compensate for the latency, coupling algorithms (FOH or EROS) are used at the input of the receiving prototype. At $t = 1$ s, a discontinuity occurs in the signal. During the following $k = 3$ macro time steps, the coupling algorithms extrapolate the signal based on its dynamics before the discontinuity until the first signal value after the discontinuity arrives at the input of the coupling algorithms. In the left part of Figure 6, where discontinuity detection is not enabled, the usage of FOH and EROS leads to significant errors, as the signal contains high-frequency components at that point. FOH amplifies the discontinuity by a factor of four for one time step, according to Equation (8). The usage of EROS results in a large extrapolation error for five consecutive time steps, as this is the duration it uses signal values for before and after the discontinuity for extrapolation, according to Equation (19).

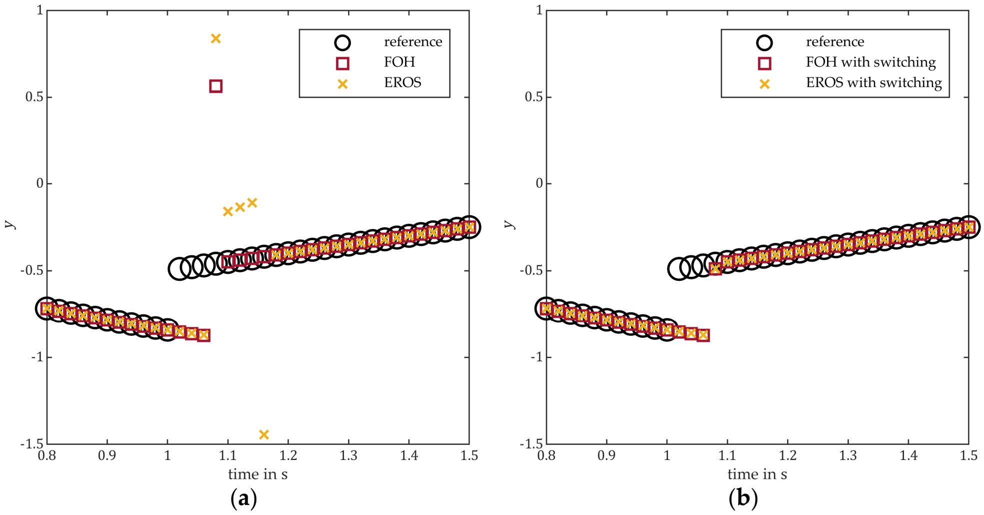


Figure 6. Behavior of coupling algorithms FOH and EROS in the case where a discontinuity occurs in the coupling signal for a latency of $k = 3$ macro steps: (**a**) without discontinuity detection; (**b**) with discontinuity detection and algorithm switching.

With enabled discontinuity detection, the extrapolation error is significantly reduced. As seen in the right part of Figure 6, the discontinuity is correctly detected, and in the subsequent time step, ZOH is used instead of the selected coupling algorithm for extrapolation. If the FOH is selected as the coupling algorithm, it switches back to FOH from the second time step after the discontinuity. In the case where EROS is used as the coupling algorithm, it uses FOH for four time steps starting from the second time step before switching back to EROS. This is an alternative to using ZOH for the entire five time steps, in which EROS is not appliable due to the detected discontinuity. The transition from FOH to EROS is not noticeable in the right part of Figure 6 because FOH and EROS yield similar results for the displayed signal. Note that the extrapolation error during the k time steps of latency (due to data transmission between the coupled prototypes) still exists when using the presented discontinuity detection. This is because the information about the occurrence of the discontinuity is only available at the input of the coupling algorithm after k time steps.

2.6. Architecture of the Overall Coupling Method

In Figure 7, the architecture of the overall coupling method for RVPs, presented in the previous sections, is depicted. It shows how the individual parts of the coupling method interact when applied to compensate for the latency of a coupling signal, y, at time t_n. The software modules are implemented using the FMI standard, as introduced in Section 1, enabling its applicability in common simulation tools where virtual prototypes are executed, as well as in test bench software that orchestrates the data exchange of real prototypes. Within the FMU, the coupling algorithm module serves as the core component. It implements the generic coupling algorithms ZOH and FOH and the novel EROS algorithm introduced in Section 2.2. Additionally, the FFNN from Section 2.3 and coupling algorithms with parameters obtained from the optimization presented in Section 2.4.2 can be selected. At any given time, one of these algorithms is active and compensates for the current latency of k, estimating the signal value $\hat{y}_n$. The necessary past signal values are provided by the signal history module. Furthermore, the FMU includes the method for detecting discontinuities in the coupling signal from Section 2.5, implemented in the corresponding module. As suggested there, it is connected to a module to temporarily switch the coupling algorithm after detecting a discontinuity to prevent large coupling errors in the signal during the macro steps that follow the discontinuity. Additionally, the latency calculator is located within the FMU, which determines the current number of macro time steps delay, k, at run time based on timing information contained within the signal messages, enabling precise latency compensation and adaptation to variable latency times.

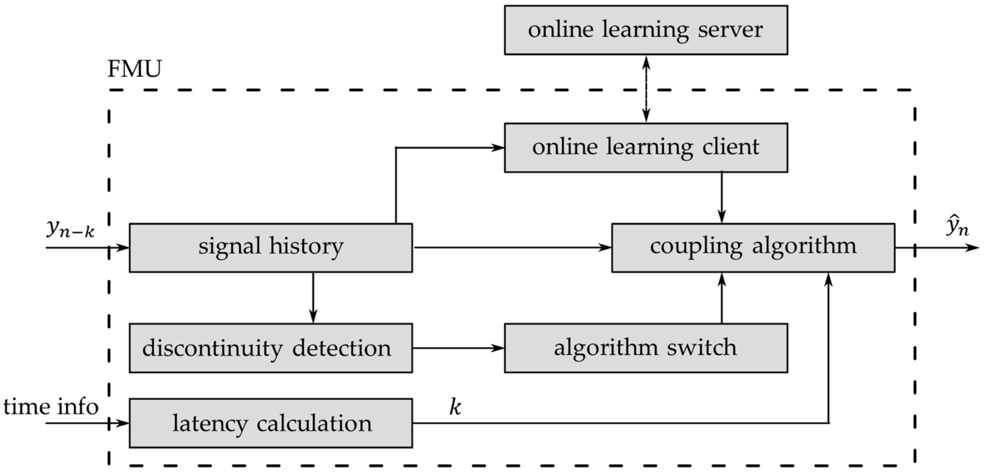


Figure 7. Software modules within the functional mock-up unit (FMU) containing the individual parts of the presented coupling method and their interconnections.

In each time step, the FMU executes a do-step method, sequentially calling the methods of the individual modules. If the current time step is a macro time step, the determination of the current latency time occurs first. Subsequently, the current signal value is examined for discontinuity, and the coupling algorithm is chosen accordingly. At the end of the do-step method, the actual latency compensation in the signal takes place by calculating the signal value $\hat{y}_n$ using the selected coupling algorithm. If the current time step is a micro time step, only the coupling algorithm is executed to determine the signal value for that micro time step by reconstructing the continuous signal. The online learning for adapting the FFNN coupling algorithm at run time is too computationally intensive to repeat in every time step, which is why it takes place outside the FMU. Thereby, the online learning client provides a past section of the coupling signal as training data to the online learning server and receives the new parameters of the FFNN after the training is completed, which are then used to update the coupling algorithm.

3. Results

3.1. Productively Used Real-Virtual Prototype of a Hybrid Vehicle

In this chapter, we demonstrate the benefits of the coupling method for latency compensation developed in Section 2 and show its practical application by using it for an RVP that is productively used at Robert Bosch GmbH for deriving operation strategies of hybrid electric vehicles. The RVP (the architecture and purpose of which are detailed by the authors of [54]) represents a passenger vehicle with a P2.5 mild hybrid topology as the propulsion system concept. In this topology, in addition to an internal combustion engine, an electric motor generator is installed on the side of the dual-clutch transmission in the vehicle, enabling both regenerative braking and periods of pure electric driving. The main advantage of mild hybrid systems is their potential to save fuel compared to purely conventional systems. For the investigated P2.5 architecture, the average potential fuel savings amount to approximately 19% [55]. Furthermore, the optimization possibilities of the hybrid operation strategy regarding the fuel consumption and propulsion energy generation of the combustion engine are within the single-digit percentage range [56]. Thus, to define optimal operating strategies based on the investigated RVPs, it is crucial that the transferred propulsion energy between the coupled prototypes is not distorted due to the network effects. Therefore, the goal of applying the developed coupling method on this RVP is to compensate for latency times and thereby minimize energy conservation violations in the coupling between coupled prototypes.

Figure 8 schematically illustrates the structure of the RVP under investigation, including the significant coupling signals between the two interconnected real prototypes and the one virtual prototype. The first real prototype is an internal combustion engine, located on an engine test bench and physically connected to a dynamometer acting as a counter load. The second is a real engine control unit (ECU) connected to the engine via a controller area network (CAN) bus. The virtual prototype comprises a simulation model that includes the vehicle dynamics, the electric motor generator, the battery, the drivetrain, including a dual-clutch transmission, and other vehicle components, such as electrical consumers. Additionally, the simulation model includes a virtual driver who controls the real-virtual vehicle according to a given speed profile. The simulation model is executed under soft real-time conditions on a simulation PC with a Windows operating system. The simulation PC is physically separated from the engine test bench and located in a different room. Via a network, it is connected to the control system of the engine test bench, which orchestrates the data exchange with the engine.

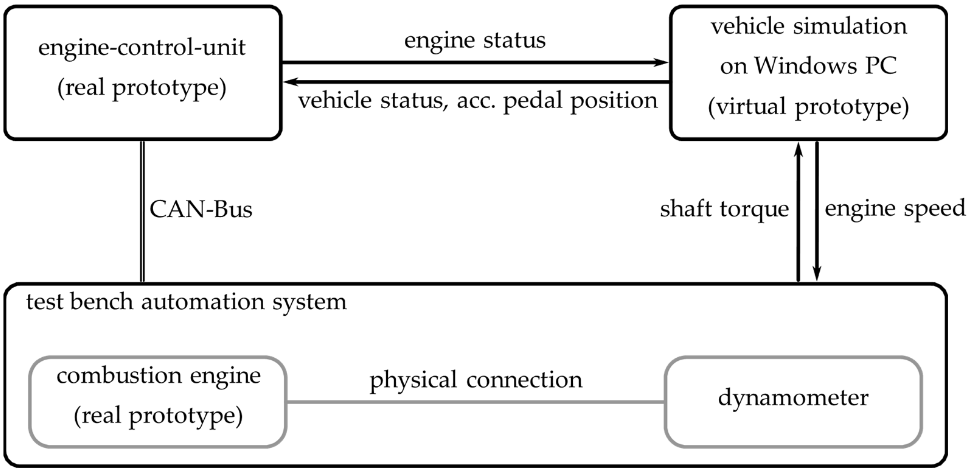


Figure 8. Structure of the utilized RVP from industry, with a focus on the data exchange, as introduced in reference [54].

3.2. Analysis of the RVP

Prior to the application of the developed coupling method, an analysis of the RVP is carried out. Firstly, the technical constraints and the resulting coupling faults of the data exchange between the coupled prototypes are identified to determine which coupling signals require latency compensation. Secondly, the parameters necessary for designing an optimal coupling algorithm are identified, as outlined in Section 2.4.2.

The coupling between the engine test bench and the simulation model is identified as the most critical, as it is a power bond interface [57] through which the entire propulsion energy from the combustion engine is transmitted. Even small coupling errors due to sampling or latency violate the energy conservation within such a power bind coupling, causing a distortion of the system's behavior. A key focus in the development of hybrid strategies, for which the investigated RVP is used, is the reduction in fuel consumption [54], which is closely related to the amount of energy transmitted through this interface. Therefore, the reduction in coupling errors at this interface is of great interest to increase the credibility of the results obtained regarding an optimal hybrid strategy (see Section 3.1). A latency compensation in the other coupling interfaces of the RVP is not necessary. Latencies in the CAN bus between the test bench and the ECU correspond to those that occur in the CAN bus during the operation of a real vehicle, which is why no compensation is required here.

The signals transmitted between the ECU and the simulation model have lower dynamics compared to those in the power bond, which means that latency times at this interface have a relatively small impact on the system behavior.

The round-trip time between the simulation and the test bench, which includes a phase delay of a low pass filter due to noise in the measurement of the output signals of the test bench, has been measured as $\tau(t) = 0.0575$ s ± 0.0005 s. Therefore, by utilizing Equation (4), the macro step size is chosen to be $\Delta T = 0.01$ s, resulting in a latency of $k\Delta T = 0.06$ s with $k = 6$ macro steps delay. This configuration ensures that the latency remains nearly constant, enabling analysis in the frequency domain, as demonstrated in Section 2.4. Moreover, due to the filter, the majority of the round-trip time is spent in the direction from the test bench to the simulation, and a small shift of the time base of those two prototypes is tolerable; the entire latency is compensated for by the shaft torque signal at the input of the simulation model (refer to Figure 8), requiring only a single instance of the coupling algorithm.

3.2.1. Frequency Domain Analysis of the Coupling Process of the Torque Signal

In order to analyze which coupling algorithms are suitable for compensating the latency in the considered shaft torque signal and to optimize a coupling algorithm based on the method from Section 2.4.2, the left part of Figure 9 presents the frequency spectrum of the signal obtained through the Fourier analysis of the signal measurements. The signal exhibits its highest magnitudes within the frequency range up to 10 rad/s, which corresponds to 3.2% of the Nyquist frequency for the given sampling with $\Delta T = 0.01$ s. Additionally, the signal is superimposed with an oscillation in the range between 30 rad/s and 55 rad/s, which is up to 17.5% of the Nyquist frequency. According to Table 1, the frequency range of the superimposed oscillation is very much above the upper limit of the frequency range in which the coupling algorithms under consideration are to be used for the given $k = 6$. The bode plot of FOH and EROS, depicted alongside others in the right part of Figure 9, confirms this, as both algorithms amplify the magnitude within the frequency range of the oscillations significantly.

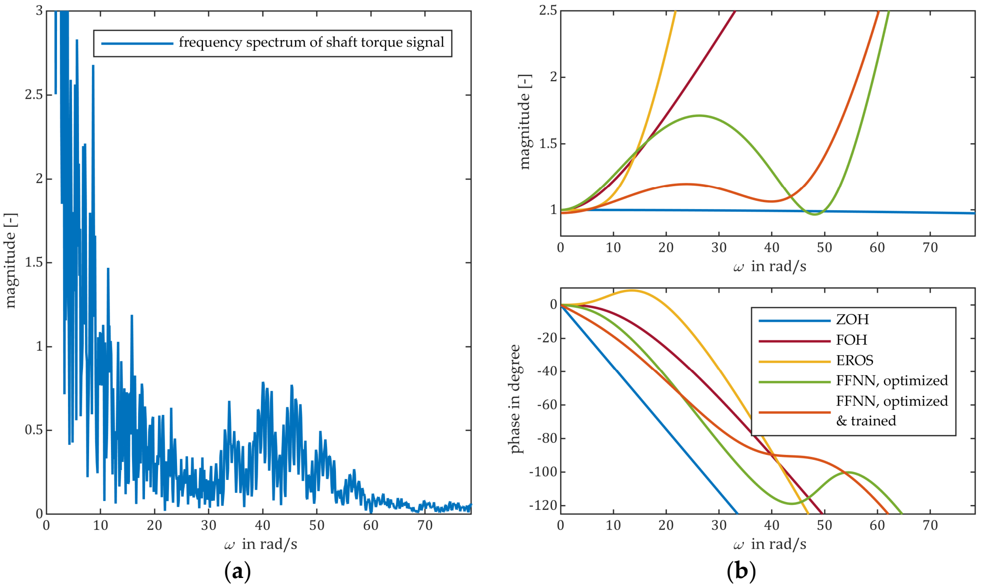


Figure 9. Analysis of RVP in the frequency domain: (**a**) Spectrum of the torque signal with $\Delta T = 0.01$, focusing on small magnitudes; (**b**) Bode plot of the optimized and learned feedforward neural network (FFNN) coupling algorithm compared to the algorithms ZOH, FOH, and EROS for the given latency of the investigated RVP defined by $k = 6$ and $\Delta T = 0.01$ s. Nyquist frequency is at 314.16 rad/s. FOH reaches its maximal magnitude amplification of 10.2 at 232.3 rad/s, EROS an amplification of 17.0 at 139.2 rad/s, FFN optimized an amplification of 11.6 at 128.6 rad/s, and FFN optimized and trained an amplification of 10.0 at 127.9 rad/s.

The effects in coupling signals, such as these superimposed high-frequency oscillations, are the reason why the naïve ZOH algorithm is commonly used as the coupling algorithm in most RVPs in the industry. By not amplifying the magnitude of the coupling signals, ZOH reduces the risk of significant coupling errors and prevents the sudden distortion of results due to unwanted amplified high frequencies in the signal. Two approaches are presented to address this issue. The first approach, as demonstrated in Section 3.3, utilizes preceding discontinuity detection, allowing for coupling algorithms, such as FOH and EROS, which may amplify certain frequencies, to be applied without the need for detailed frequency analysis of the signals. The second approach is the optimization and online adaptation of an FFNN algorithm, as illustrated in the following section.

3.2.2. Optimal and Trained FFNN Coupling Algorithms

For the optimal design of a coupling algorithm (according to Section 2.4.2), the only required system parameters are the bandwidth of the signal on which the coupling algorithm is applied and the relative degree of the coupled system. Due to the presence of the above-mentioned superimposed oscillations, different frequency ranges in terms of the magnitude and phase costs are defined within the cost function of the optimization problem. For latency compensation, a neutral phase shift is only important within the bandwidth of the signal itself, while a neutral magnitude gain must also apply to the superimposed oscillations to avoid amplifying them. Based on the spectrum of the torque signal shown in the left part of Figure 9, the frequency ranges are defined as

$$\omega_{b,p} \in \left[\omega_{l,p}, \omega_{u,p}\right] = \left[0.1\frac{\text{rad}}{\text{s}}, 10\frac{\text{rad}}{\text{s}}\right] \tag{43}$$

for the phase term and

$$\omega_{b,\text{m}} \in [\omega_{l,m}, \omega_{u,m}] = \left[0.1\frac{\text{rad}}{\text{s}}, 55\frac{\text{rad}}{\text{s}}\right] \tag{44}$$

for the magnitude term. The second needed parameter—the relative degree of the transfer behavior of the open loop consisting of the engine test bench and the vehicle simulation model—is conservatively estimated to be $r = 2$, as neither of the two systems exhibits a direct feedthrough between the input and output signals of shaft torque and engine speed. By using the weighting parameters calculated through the solution of the optimization problem, an FFNN coupling algorithm is generated using the "Network Morphism" methods introduced in Section 2.3. This algorithm has the optimized behavior and can be adapted through online learning during the execution of the RVP. The algorithm utilizes $m = 9$ past signal value as input, as this is the smallest value at which the result of the optimization problem no longer significantly improves.

The transfer behavior of the optimized FFNN coupling algorithm is also shown in the right part of Figure 9 above. As specified in the cost function, the magnitude gain of the optimized FFNN algorithm is significantly lower in the range of the superimposed oscillations compared to EROS and FOH. However, for low frequencies, the phase error and, thus, its ability to compensate for latency, is worse compared to EROS.

The other transfer function shown in Figure 9 represents the same FFNN coupling algorithm after its adaptation through online learning during the execution of the RVP. For this purpose, two learning cycles with 100 epochs each were performed during the execution of the RVP. Compared to the optimized FFNN coupling algorithm, the additionally trained FFNN algorithm has a lower magnitude throughout the entire frequency range, including superimposed oscillations. However, this leads to a slightly higher phase error for low frequencies compared to the other non-naïve algorithms. Nonetheless, this increase is still significantly lower than for the naïve ZOH algorithm.

3.3. Results of the Coupling Method

The comparison of different configurations of the developed coupling method was performed using an acceleration maneuver that lasted 12 s for the hybrid vehicle. During the acceleration, which is briefly interrupted by a gear shift at $t = 3.2$ s, the combustion engine provides the main driving energy.

3.3.1. Preceding Investigation Using MIL Simulation

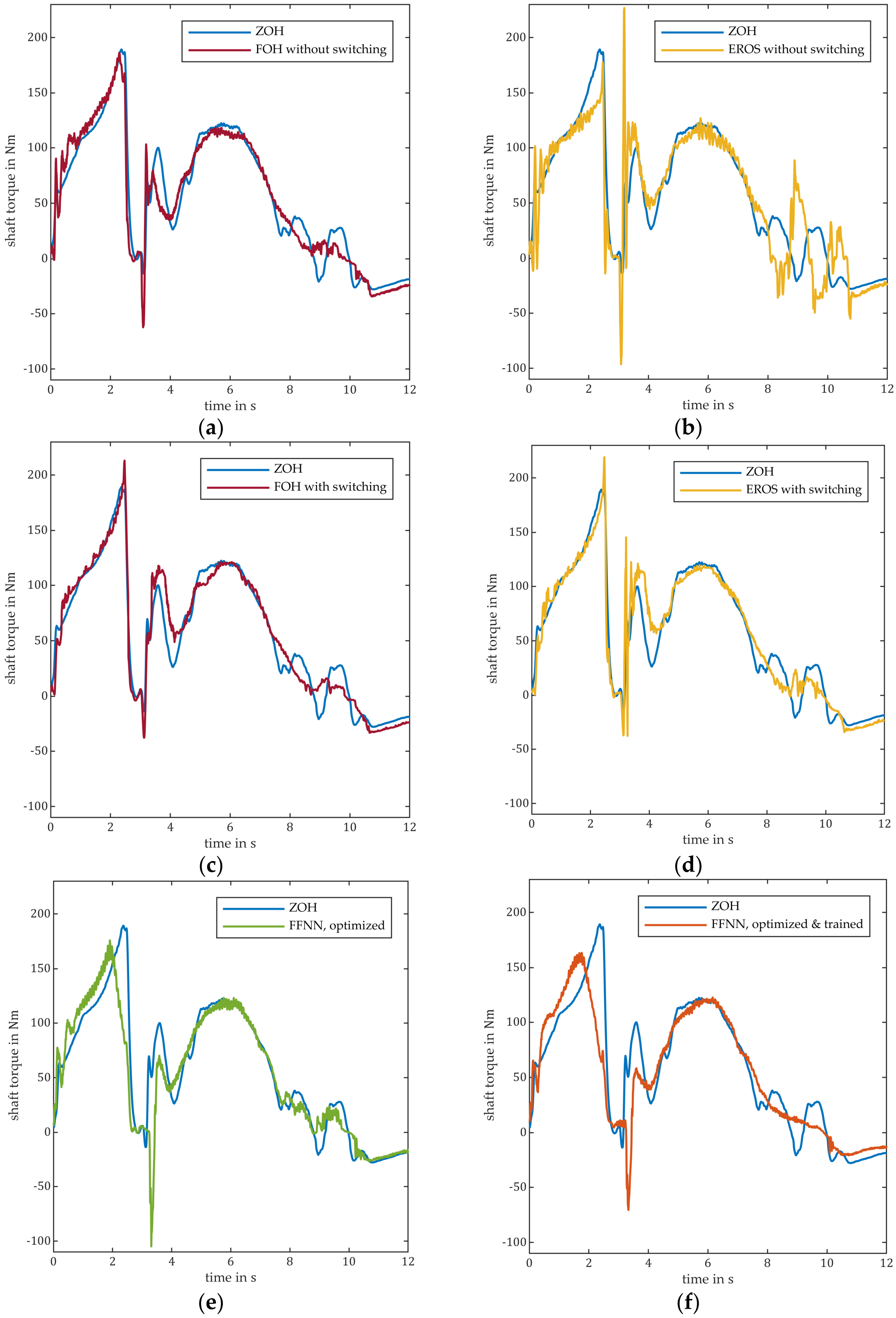
Before applying the coupling method to the RVP, an investigation is conducted to examine how a latency of $k\Delta T = 0.06$ s in the torque signal can affect the overall system behavior of the RVP, particularly the energy consumption of the combustion engine, which is a critical factor for optimizing hybrid strategies using the RVP. In order to examine this, a purely virtualized version of the RVP is utilized, where all real prototypes are available as simulation models. The impact of latency on the system behavior of the RVP is quantified by comparing simulations with an idealized coupling without latency between the prototypes and simulations, where the latency is accurately modeled according to the real latency in the RVP. It is calculated from the shaft torque and engine speed signals (see Figure 8) in that, in the absence of latency and during the considered acceleration maneuver, there is 2.5% less energy transferred from the combustion engine compared to the simulation with latencies, while the distance traveled remains the same. This difference in transmitted energy due to violations of the energy conservation at the coupling interface [57] is significant when considering that the overall optimization possibility regarding energy consumption in the mild hybrid system is within the single-digit percentage range (see Section 3.1). Additionally, the simulation results show that the first peak in the torque signal occurs 0.3 s earlier, when no latency is present.

3.3.2. Results for the RVP

In order to evaluate the effectiveness of the different configurations of the coupling method, the acceleration maneuver was run multiple times with the RVP of the hybrid vehicle. Figure 10 shows the torque signal for various configurations at the output of the FMU, which implements the entire coupling method, as shown in Section 2.6. Each configuration was compared to the naïve ZOH algorithm, where the coupling signal remains unchanged, and no latency compensation takes place. Based on the preceding analysis using simulation (see Section 3.3.1), it can be inferred that a good latency compensation requires an earlier occurrence of the first-moment peak compared to ZOH. In addition, with good latency compensation, deviations in the signal course are to be expected compared to the signal course when using ZOH, as the amount of energy transferred from the combustion engine to the test bench must change.

In Figure 10, an amplification of high frequencies for FOH and EROS is observed when no preceding discontinuity detection and algorithm switching is used. This occurs specifically when the frequencies in the signal are high.

For instance, it is observed at the beginning of the acceleration, during the gear shift at $t = 3.2$ s, and in the presence of the superimposed oscillations at around $t = 6$ s, as identified in the frequency spectrum in Figure 9. By using the discontinuity detection mechanism and switching the algorithm after a positive detection, as proposed in Section 2.5, the amplification of high frequencies is significantly reduced. Additionally, it is observed that, in this case, the output of the coupling algorithm is very similar when selecting FOH or EROS as the primary algorithm. The most significant qualitative differences in the behavior, compared to ZOH, were observed when using the FFNNs. Particularly, during the first acceleration phase, at around $t = 2$ s, the peak of the torque signal is reached earlier and is lower compared to when using the ZOH algorithm. The superimposed oscillations are slightly amplified when using the FFNN that has only been optimized, but less amplified when the additionally trained FFNN is chosen as the coupling algorithm.



(**a**) (**b**)

(**c**) (**d**)

(**e**) (**f**)

Figure 10. Latency-compensated shaft torque signal of the RVP when using different coupling algorithms, as compared to the naïve ZOH method: (**a**,**b**) FOH and EROS, without switching the algorithm; (**c**,**d**) FOH and EROS with switching the algorithm after a detected discontinuity; (**e**) FFNN optimized using the method from Section 2.4.2; (**f**) FFNN initialized, like the FFNN in (**e**) that was trained online during a previous run of the RVP.

No discontinuity detection is used together with the FFNNs here, as the frequency range of the superimposed oscillations is considered in the optimization and in the training data of the FFNN algorithm. Thus, the algorithms are optimized for this frequency range, and switching the algorithm is not necessary.

For an additional quantitative evaluation of the results obtained using the investigated configurations of the coupling method, the coupling error defined in Equation (5) is calculated using the metric proposed by Sprague and Geers [58] as the distance function. The metric consists of three values and distinguishes between magnitude error $M_{S\&G}$, phase error $P_{S\&G}$, and the combination $C_{S\&G}$ of both, allowing the user to draw a conclusion on the cause of the coupling error in addition to its quantification. According to Sprague and Geers, it holds that

$$M_{S\&G} = \sqrt{\frac{\sum_{i=1}^{N} {y_i}^2}{\sum_{i=1}^{N} \hat{y}_i^2}} - 1,\ P_{S\&G} = \frac{1}{\pi}\cos^{-1}\left(\frac{\sum_{i=1}^{N} y_i \hat{y}_i}{\sqrt{\sum_{i=1}^{N} y_i^2 \sum_{i=1}^{N} \hat{y}_i^2}}\right) \text{ and } C_{S\&G} = \sqrt{M_{S\&G}^2 + P_{S\&G}^2}, \tag{45}$$

where $\boldsymbol{Y}$ is the vector of N signal values, y_i, at macro time points of a coupling signal at the output of a prototype, and $\hat{\boldsymbol{Y}}$ is the corresponding vector at the output of the coupling algorithm at the input of another prototype.

The three parts of the coupling error for the different configurations shown in Figure 10 above are listed in Table 2. Compared to the naïve ZOH algorithm, the combined coupling error is reduced by 35% and 40% when using the optimized and additionally trained FFNNs, respectively. Notably, the phase error, which is the cause of almost all the coupling errors when using ZOH, is significantly reduced when using the FFNNs and experiences only a slight increase in the magnitude error. Without the use of discontinuity detection and algorithm switching, both FOH and particularly the EROS algorithm result in an increased error compared to ZOH. However, with preceding discontinuity detection algorithm switching, an improvement of 7% in the combined coupling error is observed for both algorithms.

Table 2. Summarized coupling error using the metric by Sprague and Geers as a distance measure for the investigated coupling algorithms.

	$C_{S\&G} \cdot 100$	$M_{S\&G} \cdot 100$	$P_{S\&G} \cdot 100$
ZOH	4.53	0.01	4.53
FOH without switching	4.43	0.86	4.35
EROS without switching	6.54	2.12	6.18
FOH with switching	4.24	0.79	4.17
EROS with switching	4.21	0.88	4.11
FFNN, optimized	2.94	0.65	2.87
FFNN, trained	2.71	0.67	2.62

Furthermore, Table 3 presents the two values identified in the simulation in Section 3.3.1, which quantify the impact of latency on the behavior of the system for the different coupling method configurations analyzed. While the values for FOH, EROS with and without discontinuity detection, and algorithm switching show only minor changes compared to ZOH, the changes for the FFNNs are larger and are close to the values determined for ideal latency compensation using simulation. The energy conservation violations due to the coupling faults are compensated for nearly entirely when using the trained FFNN coupling algorithm.

Table 3. Values representing the system behavior of the RVP for the investigated coupling algorithms compared to ZOH. For comparison, the values obtained from the simulation of the MIL simulation of the RVP are listed at the bottom. There, the difference between a simulation with ideal coupling and a simulation with simulated coupling faults compensated with the naïve ZOH method is shown.

	Time of First Torque Peak with Respect to ZOH in s	**Energy Transferred with Respect to ZOH in %**
FOH without switching	−0.07	−0.62
EROS without switching	0.10	−0.22
FOH with switching	0.12	−0.09
EROS with switching	0.12	−1.07
FFNN, optimized	−0.42	−4.39
FFNN, trained	−0.38	−2.92
Ideal coupling from MIL simulation of RVP	−0.30	−2.50

3.4. Discussion of the Experimental Results

The technical constraints on the investigated RVP of a hybrid vehicle are challenging for latency compensation. Firstly, the round-trip time of 0.06 s is quite high for a physical power bond interface in a highly dynamic system. Secondly, the useful signal is superimposed with oscillations in the double-digit percentage range of the Nyquist frequency. However, the results from the previous section demonstrate that the coupling method presented in this contribution is applicable to RVPs and improves their system behavior.

The optimized coupling algorithm, along with the additionally trained FFNN algorithm, finds a balance between compensating for latency in the useful signal and minimizing the amplification of the superimposed frequencies. As a result, they have been proven to effectively reduce the coupling error by up to 40%, as well as the energy conservation violations in the critical coupling interface between a combustion engine and the vehicle simulation of the RVP of a hybrid electric vehicle. This improves the credibility of the hybrid operating strategies that are derived using this RVP, as optimization possibilities of the hybrid operation strategy are within the single-digit percentage range (see Section 3.1).

Therefore, the applicability of the method for the optimal design of a coupling algorithm based on the frequency domain analysis of the coupling process from Section 2.4.2 on an RVP is confirmed. The limited system knowledge required for the design can be derived with minimal effort from measurements of the coupling signals. The resulting coupling algorithm significantly reduces coupling errors under challenging technical conditions, outperforming all investigated generic algorithms. The design of an FFNN coupling algorithm based on the optimization results and its online learning is also successfully applicable to the RVP. Since the trained FFNN has a higher number of neurons and, thus, more parameters available than those used for optimal design, it adapts even better to the coupling signal. As a result, compared to the purely optimized algorithm, both the coupling error and the overall system behavior of the RVP are further improved.

The discontinuity detection, including the switching of coupling algorithms, proves to be useful as well, although the resulting reductions in coupling error are lower compared to when using FFNNs. It allows for the use of generic coupling algorithms for coupling signals, the frequencies of which are not always within the frequency range that the algorithms are to be used within, e.g., due to discontinuities or occasional high frequencies caused by superimposed oscillations, as in the investigated RVP. Consequently, the application range of generic algorithms expands, allowing for their use even in scenarios where prior knowledge of the frequencies present in the coupling signal is unavailable.

The advantage of the proposed generic EROS coupling algorithm over FOH is not as clear as the analysis in the frequency domain suggests by the good balance between the magnitude and phase errors of EROS (refer to Figure 3). This can be explained by the fact that even when EROS is selected as a coupling algorithm, it is not actually always utilized because the discontinuity detection switches to ZOH or FOH due to the presence of high frequencies in major parts of the coupling signal.

4. Conclusions and Future Work

We presented a coupling method for real-time co-simulation, which contributes to the utilization of detailed virtual prototypes in the later stages of the development process for cross-domain mechatronic systems. The method improves the coupling of virtual prototypes with real hardware components, so-called RVPs, by compensating for latencies that are identified as the major reason for coupling faults due to network communication. The core of the method, developed with a focus on simplicity of application, is the coupling algorithm for which we presented two alternatives. The first is a novel generic algorithm called EROS that is derived from a developed procedure of extrapolation in the error space. Additionally, we demonstrated the design of an optimal coupling algorithm based on an analysis of the coupling process of RVPs in the frequency domain, requiring minimal system knowledge of the RVP under investigation. Furthermore, we showed that the optimized algorithm, as well as EROS and other known generic algorithms, can be transformed into an FFNN coupling algorithm. Starting from a good initial parametrization, the FFNN algorithm allows for adaption to a coupling signal through online training with limited computation effort.

By introducing preceding discontinuity detection with coupling algorithm switching, we contributed to expanding the application range of generic coupling algorithms, as the usage of the discontinuity detection reduces the risk of amplifying occasional high frequencies in the coupling signal. In our opinion, the risk of amplifying high frequencies, which may significantly increase the coupling error of RVPs, is the main reason why latency compensation is often omitted in industrial RVPs, and the naïve ZOH coupling algorithm is used instead.

We confirmed the effectiveness of the method by applying all parts of the method to an RVP of a hybrid vehicle, which is actively used in industrial development projects for determining hybrid operating strategies. Particularly, the FFNN coupling algorithm resulting from a combination of optimization based on existing system knowledge and subsequent training showed a 40% reduction in coupling error, positively influencing the system behavior of the RVPs. The entire coupling method is implemented using the FMU standard, making it easily applicable in most simulation and test automation software tools. The developed method is not limited or specialized to application in the automotive industry but can be applied to RVPs in any industry.

Future work can expand upon this research by further investigating the presented error space extrapolation method and utilizing it to derive potentially improved generic coupling algorithms. Additionally, the analysis method of the coupling process in the frequency domain should be expanded to enable the analysis of larger prototype networks with a greater number of coupling signals, making it applicable to a wider range of RVPs. Furthermore, some of the methods presented here, such as the EROS coupling algorithm or discontinuity detection with algorithm switching, may also be useful for non-iterative offline co-simulation, which should be further explored.

5. Patents

A patent and a patent application have resulted from the presented work. The patent titled “Method and device for synchronizing a simulation with a real-time system” [49] protects the method mentioned in Section 2.4.1 for aligning sampling sequences of virtual and real prototypes to reduce or harmonize the latency of data transmission between them. Additionally, a patent application was filed on 09 October 2023 for the method presented in Section 2.2, which involves constructing coupling algorithms in an error space.

Author Contributions: Conceptualization, P.B. and L.M.; methodology, P.B.; software, P.B. and O.K.; validation, P.B., O.K. and L.M.; investigation, P.B.; writing—original draft preparation, P.B.; writing—review and editing, O.K., L.M. and D.S.; funding acquisition, D.S.; supervision, O.K., L.M. and D.S. All authors have read and agreed to the published version of the manuscript.

Funding: We acknowledge support by the Open Access Publication Fund of the University of Duisburg-Essen.

Data Availability Statement: The software and data utilized in this research are the property of Robert Bosch GmbH. Access to the software and data can be obtained from the authors, subject to the permission of Robert Bosch GmbH.

Conflicts of Interest: Peter Baumann and Oliver Kotte disclose that they are employees at Robert Bosch GmbH and that they are the inventors of the pending patent mentioned in Section 5. Together with Lars Mikelsons, they are also the inventors of the other patent [49] mentioned in Section 5. The remaining authors declare that the research was conducted in the absence of any other commercial or financial relationships that could be construed as a potential conflict of interest.

Symbols

t	Time in s	α	Factor of LRELU activation function
ΔT	Macro time step in s	$\boldsymbol{x}$	Vector between FFNN layers
k	Number of macro steps latency	$G(\cdot)$	Transfer function
τ	Latency in s	$\delta(\cdot)$	Dirac function
u	Input signal in time domain	U	Input signal in frequency domain
$y, \hat{y}$	Output signal in time domain	$Y,\ \hat{Y}$	Output signal in frequency domain
$\boldsymbol{a},\ \boldsymbol{A}$	Weighting parameters of a coupling algorithm	s	Complex frequency domain parameter
m	Number of signal values used by coupling algorithm	ω	Angular frequency in rad/s
e	Coupling error	J	Cost function
$d(\cdot)$	Distance function	r	Relative degree
$f_p(\cdot)$	Set of curves	j	Imaginary unit
$\mathbf{W}$	Weight matrix of neuron	$h(\cdot)$	Window function
$\mathbf{\Theta}$	Offset vector of neuron	ξ	Parameter of discontinuity detection
$\sigma(\cdot)$	Activation function of neuron	$C_{S\&G},\ M_{S\&G},\ P_{S\&G}$	Coupling error as defined by Sprague and Geers

References

1. Mina, A.A.; Braha, D.; Bar-Yam, Y. Complex engineered systems: A new paradigm. In *Complex Engineered Systems: Science Meets Technology*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 1–21.
2. VDI/VDE 2206. *Entwicklung Mechatronischer und Cyber-Physischer Systeme*; Beuth Verlag GmbH: Düsseldorf, Germany, 2021.
3. Gomes, C.; Thule, C.; Larsen, P.G.; Denil, J.; Vangheluwe, H. Co-Simulation of Continuous Systems: A Tutorial. 2018. Available online: https://arxiv.org/pdf/1809.08463.pdf (accessed on 8 February 2024).
4. Blochwitz, T.; Otter, M.; Akesson, J.; Arnold, M.; Clauß, C.; Elmqvist, H.; Friedrich, M.; Junghanns, A.; Mauß, J.; Neumerkel, D.; et al. Functional Mockup Interface 2.0: The Standard for Tool independent Exchange of Simulation Models. In Proceedings of the 9th International MODELICA Conference, Munich, Germany, 3–5 September 2012; Linköping University Electronic Press: Linköping, Sweden, 2012; pp. 173–184, ISBN 1650-3740.
5. Hansen, S.T.; Gomes, C.Â.G.; Najafi, M.; Sommer, T.; Blesken, M.; Zacharias, I.; Kotte, O.; Mai, P.R.; Schuch, K.; Wernersson, K. The FMI 3.0 Standard Interface for Clocked and Scheduled Simulations. *Electronics* **2022**, *11*, 3635. [CrossRef]
6. Boumans, M.; Johannaber, M.; Schulmeister, U. Consistent application of systems engineering and simulation for cross-domain function integration. In Proceedings of the 19. Internationales Stuttgarter Symposium: Automobil-und Motorentechnik, Stuttgart, Germany, 19–20 March 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 508–522, ISBN 3658259388.
7. Eilers, S.; Müller-Schloer, C. Mixed Virtual/Real Prototypes for Incremental System Design—A Proof of Concept. In *Embedded Computer Systems: Architectures, Modeling, and Simulation*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 465–474. ISBN 978-3-540-31664-0.

8. Mihalič, F.; Truntič, M.; Hren, A. Hardware-in-the-loop simulations: A historical overview of engineering challenges. *Electronics* **2022**, *11*, 2462. [CrossRef]
9. Szalay, Z. Next generation X-in-the-loop validation methodology for automated vehicle systems. *IEEE Access* **2021**, *9*, 35616–35632. [CrossRef]
10. Klein, S.; Savelsberg, R.; Xia, F.; Guse, D.; Andert, J.; Blochwitz, T.; Bellanger, C.; Walter, S.; Beringer, S.; Jochheim, J.; et al. Engine in the Loop: Closed Loop Test Bench Control with Real-Time Simulation. *SAE Int. J. Commer. Veh.* **2017**, *10*, 95–105. [CrossRef]
11. Steinbrink, C.; Lehnhoff, S.; Rohjans, S.; Strasser, T.I.; Widl, E.; Moyo, C.; Lauss, G.; Lehfuss, F.; Faschang, M.; Palensky, P. Simulation-based validation of smart grids–status quo and future research trends. In Proceedings of the International Conference on Industrial Applications of Holonic and Multi-Agent Systems, Lyon, France, 28–30 August 2017; Springer: Berlin/Heidelberg, Germany, 2017; pp. 171–185.
12. Sadjina, S.; Kyllingstad, L.T.; Rindarøy, M.; Skjong, S.; Æsøy, V.; Pedersen, E. Distributed co-simulation of maritime systems and operations. *J. Offshore Mech. Arct. Eng.* **2019**, *141*, 11302. [CrossRef]
13. Baumgartner, E.; Ronellenfitsch, A.; Reuss, H.-C.; Schramm, D. Using a dynamic driving simulator for perception-based powertrain development. *Transp. Res. Part F Traffic Psychol. Behav.* **2019**, *61*, 281–290. [CrossRef]
14. Ivanov, V.; Augsburg, K.; Bernad, C.; Dhaens, M.; Dutré, M.; Gramstat, S.; Magnin, P.; Schreiber, V.; Skrt, U.; van Kelecom, N. Connected and Shared X-in-the-Loop Technologies for Electric Vehicle Design. *World Electr. Veh. J.* **2019**, *10*, 83. [CrossRef]
15. Alfonso, J.; Rodriguez, J.M.; Bernad, C.; Beliautsou, V.; Ivanov, V.; Castellanos, J.A. Geographically distributed real-time co-simulation of electric vehicle. In Proceedings of the 2022 8th International Conference on Control, Decision and Information Technologies (CoDIT), Istanbul, Turkey, 17–20 May 2022; pp. 1002–1007.
16. The Modelica Association. Distributed Co-Simulation Protocol (DCP). The Modelica Association. 2019 (1.0.0). Available online: https://dcp-standard.org/ (accessed on 8 February 2024).
17. Krammer, M.; Schiffer, C.; Benedikt, M. ProMECoS: A process model for efficient standard-driven distributed co-simulation. *Electronics* **2021**, *10*, 633. [CrossRef]
18. Krammer, M.; Benedikt, M.; Blochwitz, T.; Alekeish, K.; Amringer, N.; Kater, C.; Materne, S.; Ruvalcaba, R.; Schuch, K.; Zehetner, J.; et al. The Distributed Co-Simulation Protocol for the Integration of Real-Time Systems and Simulation Environments. In Proceedings of the 50th Computer Simulation Conference, Bordeaux, France, 9–12 July 2018; Society for Computer Simulation International: San Diego, CA, USA, 2018.
19. Segura, M.; Calderón, A.J.; Poggi, T.; Barcena, R. Testing the Verification and Validation Capability of a DCP-Based Interface for Distributed Real-Time Applications. *Electronics* **2023**, *12*, 4919. [CrossRef]
20. Rautenberg, P.; Weber, P.; Degel, J.P.; Hähnlein, S.; Gauterin, F.; Koch, T.; Doppelbauer, M.; Gohl, M. Electrified Powertrain Development: Distributed Co-Simulation Protocol Extension for Coupled Test Bench Operations. *Appl. Sci.* **2023**, *13*, 2657. [CrossRef]
21. Meyer, M.-A.; Sauter, L.; Granrath, C.; Hadj-Amor, H.; Andert, J. Simulator coupled with distributed co-simulation protocol for automated driving tests. *Automot. Innov.* **2021**, *4*, 373–389. [CrossRef]
22. Baumann, P.; Krammer, M.; Driussi, M.; Mikelsons, L.; Zehetner, J.; Mair, W.; Schramm, D. Using the Distributed Co-Simulation Protocol for a Mixed Real-Virtual Prototype. In Proceedings of the 2019 IEEE International Conference on Mechatronics, Ilmenau, Germany, 18–20 March 2019; IEEE Industrial Electronics Society: Ilmenau, Germany, 2019.
23. Gan, C.; Todd, R.; Apsley, J. Mitigating time delays: An evaluation of their impact using a simulation model of an aircraft power system demonstrator facility. *IEEE Ind. Appl. Mag.* **2014**, *21*, 44–53. [CrossRef]
24. Cale, J.L.; Johnson, B.B.; Dall'Anese, E.; Young, P.M.; Duggan, G.; Bedge, P.A.; Zimmerle, D.; Holton, L. Mitigating communication delays in remotely connected hardware-in-the-loop experiments. *IEEE Trans. Ind. Electron.* **2018**, *65*, 9739–9748. [CrossRef]
25. Holiš, R.; Bobál, V.; Vojtěšek, J. Real-time digital control of time-delay systems: From smith predictor to MPC. In Proceedings of the International Conference on Engineering, Technology and Innovation (ICE/ITMC), Madeira, Portugal, 27–29 June 2017; pp. 254–263, ISBN 1538607743.
26. Baumann, P.; Mikelsons, L.; Kotte, O.; Schramm, D. Analyzing the coupling process of distributed mixed real-virtual prototypes. In Proceedings of the 33rd Annual European Simulation and Modelling Conference, ESM 2019, Palma de Mallorca, Spain, 28–30 October 2019.
27. Thummerer, T.; Stoljar, J.; Mikelsons, L. NeuralFMU: Presenting a workflow for integrating hybrid neuralODEs into real-world applications. *Electronics* **2022**, *11*, 3202. [CrossRef]
28. Stettinger, G.; Benedikt, M.; Tranninger, M.; Horn, M.; Zehetner, J. Recursive FIR-Filter design for fault-tolerant real-time co-simulation. In Proceedings of the 25th Mediterranean Conference on Control and Automation (MED), Valletta, Malta, 3–6 July 2017; pp. 461–466, ISBN 1509045333.
29. Ersal, T.; Brudnak, M.; Salvi, A.; Kim, Y.; Siegel, J.B.; Stein, J.L. An iterative learning control approach to improving fidelity in internet-distributed hardware-in-the-loop simulation. *J. Dyn. Syst. Meas. Control* **2014**, *136*, 61012. [CrossRef]
30. Stettinger, G.; Benedikt, M.; Horn, M.; Zehetner, J.; Giebenhain, C. Control of a magnetic levitation system with communication imperfections: A model-based coupling approach. *Control Eng. Pract.* **2017**, *58*, 161–170. [CrossRef]
31. Tranninger, M.; Stettinger, G.; Benedikt, M.; Horn, M. Diagnosis of interconnected systems via well tuned model-based coupling algorithms. *IFAC-PapersOnLine* **2018**, *51*, 1271–1277. [CrossRef]

32. Tandon, A.; Brudnak, M.J.; Stein, J.L.; Ersal, T. An observer based framework to improve fidelity in internet-distributed hardware-in-the-loop simulations. In Proceedings of the ASME 2013 Dynamic Systems and Control Conference, Palo Alto, CA, USA, 21–23 October 2013; American Society of Mechanical Engineers Digital Collection: New York, NY, USA, 2013.
33. Zheng, Y.; Brudnak, M.J.; Jayakumar, P.; Stein, J.L.; Ersal, T. A Predictor-Based Framework for Delay Compensation in Networked Closed-Loop Systems. *IEEE/ASME Trans. Mechatron.* **2018**, *23*, 2482–2493. [CrossRef]
34. Stettinger, G.; Zehetner, J.; Benedikt, M.; Thek, N. Extending Co-Simulation to the Real-Time Domain. In Proceedings of the SAE 2013 World Congress & Exhibition, Warrendale, PA, USA, 16 April 2013.
35. Schreiber, V.; Ivanov, V.; Augsburg, K.; Noack, M.; Shyrokau, B.; Sandu, C.; Els, P.S. Shared and Distributed X-in-the-Loop Tests for Automotive Systems: Feasibility Study. *IEEE Access* **2018**, *6*, 4017–4026. [CrossRef]
36. Tranninger, M.; Haid, T.; Stettinger, G.; Benedikt, M.; Horn, M. Fault-tolerant coupling of real-time systems: A case study. In Proceedings of the 3rd Conference on Control and Fault-Tolerant Systems (SysTol), Barcelona, Spain, 7–9 September 2016; pp. 756–762, ISBN 1509006583.
37. Benedikt, M.; Watzenig, D.; Hofer, A. Modelling and analysis of the non-iterative coupling process for co-simulation. *Math. Comput. Model. Dyn. Syst.* **2013**, *19*, 451–470. [CrossRef]
38. Frasheri, M.; Ejersbo, H.; Thule, C.; Gomes, C.; Kvistgaard, J.L.; Larsen, P.G.; Esterle, L. Addressing time discrepancy between digital and physical twins. *Robot. Auton. Syst.* **2023**, *161*, 104347. [CrossRef]
39. Box, G.E.P.; Jenkins, G.M. *Time Series Analysis: Forecasting and Control*; Holden-Day: San Francisco, CA, USA, 1970.
40. Saffo, P. Six rules for effective forecasting. *Harv. Bus. Rev.* **2007**, *85*, 122.
41. Svozil, D.; Kvasnicka, V.; Pospichal, J. Introduction to multi-layer feed-forward neural networks. *Chemom. Intell. Lab. Syst.* **1997**, *39*, 43–62. [CrossRef]
42. Dorffner, G. Neural networks for time series processing. *Neural Netw. World* **1996**, *6*, 447–468.
43. Wei, T.; Wang, C.; Rui, Y.; Chen, C.W. Network morphism. *Int. Conf. Mach. Learn.* **2016**, *48*, 564–572.
44. Chen, T.; Goodfellow, I.; Shlens, J. Net2net: Accelerating learning via knowledge transfer. *arXiv* **2015**, arXiv:1511.05641.
45. Maas, A.L.; Hannun, A.Y.; Ng, A.Y. Rectifier nonlinearities improve neural network acoustic models. *Proc. Icml* **2018**, *30*, 3.
46. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
47. Oppenheim, A.V.; Willsky, A.S.; Nawab, S.H. *Signals & Systems*, 2nd ed.; Prentice-Hall, Inc.: Upper Saddle River, NJ, USA, 1996; ISBN 0138147574.
48. Doyle, J.C.; Francis, B.A.; Tannenbaum, A.R. *Feedback Control Theory*; Courier Corporation: North Chelmsford, MA, USA, 2013; ISBN 0486318338.
49. Mikelsons, L.; Baumann, M.; Kotte, O.; Baumann, P. Method and Device for Synchronizing a Simulation with a Real-Time System. U.S. Patent 11,610,034 B2, 21 March 2023.
50. Briggs, W.L.; van Henson, E. *The DFT: An Owner's Manual for the Discrete Fourier Transform*; SIAM: Philadelphia, PA, USA, 1995; ISBN 0898713420.
51. Brandt, A. *Noise and Vibration Analysis: Signal Analysis and Experimental Procedures*; John Wiley & Sons: Hoboken, NJ, USA, 2023; ISBN 1118962184.
52. Oppenheim, A.V. *Discrete-Time Signal Processing*; Pearson Education: Bangalore, India, 1999; ISBN 8131704920.
53. Ben Khaled-El Feki, A.; Duval, L.; Faure, C.; Simon, D.; Ben Gaid, M. CHOPtrey: Contextual online polynomial extrapolation for enhanced multi-core co-simulation of complex systems. *Simulation* **2017**, *93*, 185–200. [CrossRef]
54. Lachenmaier, S.; Cross, L.; Ferrara, C.; Greis, A.; Wüst, M.; Naber, D. Virtual powertrain–Vehicle simulation on the engine test bench with an implemented P2 topology. In *20. Internationales Stuttgarter Symposium: Automobil-und Motorentechnik*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 377–392. ISBN 3658309946.
55. Melaika, M.; Mamikoglu, S.; Dahlander, P. *48 V Mild-Hybrid Architecture Types, Fuels and Power Levels Needed to Achieve 75 g CO_2/km*; SAE Technical Paper 2019-01-0366; SAE: Warrendale, PA, USA, 2019.
56. Benajes, J.; García, A.; Monsalve-Serrano, J.; Martínez-Boggio, S. Optimization of the parallel and mild hybrid vehicle platforms operating under conventional and advanced combustion modes. *Energy Convers. Manag.* **2019**, *190*, 73–90. [CrossRef]
57. Sadjina, S.; Pedersen, E. Energy conservation and coupling error reduction in non-iterative co-simulations. *arXiv* **2016**, arXiv:1606.05168. [CrossRef]
58. Sprague, M.A.; Geers, T.L. A spectral-element method for modelling cavitation in transient fluid–structure interaction. *Int. J. Numer. Methods Eng.* **2004**, *60*, 2467–2499. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.