

Predictive Hold with Error Correction Techniques that Maintain Signal Continuity in Co-Simulation Environments

Jason Wells, Maher Hasan and Charles Lucas
PC Krause and Associates

ABSTRACT

The combination of increasing performance demands, increasing system complexity, and the need for reduced program development schedule and budget costs in the aerospace industry is driving engineers to increasingly rely upon modeling, simulation, and analysis (MS&A) in the platform development cycle. One approach to ensuring that such integrated system simulations remain computationally tractable is co-simulation utilizing technology found in commercially available packages, such as PC Krause and Associates, Inc.'s (PCKA's) Distributed Heterogeneous Simulation (DHS) / FastSim software. In such co-simulation environments, dynamic models are executed in independent model spaces, with coupling between subsystems achieved by exchanging a minimal set of required data typically found at subsystem boundaries. In such environments, an important challenge that must be overcome is the estimation of communicated signals whose true values may only be updated at some reduced rate compared to the model in which they are used. This estimation has traditionally been achieved with simple sample/hold techniques, wherein the communicated signals are sampled at the discrete communication intervals, and interpolation (most often utilizing polynomial functions) is used to reconstruct the signal in between samples. While conceptually simple, this approach often gives rise to step discontinuities in the communicated signals due to the difference between the interpolating function used and the signal itself. In certain models, such step discontinuities are undesirable, as they may give rise to spurious high-frequency dynamics, resulting in slower simulation speeds and potentially inaccurate solutions.

In this paper, the problem of signal estimation and reconstruction is recast in a generalized framework based on notions of prediction and error correction. Abstracting the problem in this form allows for describing different signal estimation schemes in a unified manner, as application of different prediction and error correction formulas, including the sample/hold techniques commonly used in co-simulation tools. Based on this framework, a new prediction and error correction scheme is derived which allows for maintaining various levels of continuity in the estimated signal and in its derivatives, ensuring that smoothness is retained and step discontinuities associated with traditional sample/hold approaches are eliminated. This paper presents a detailed comparison of the proposed technique against traditional approaches, whereby it is shown that the new method yields more desirable performance under certain conditions. Ultimately, the generalized framework and the proposed approach put forth in this paper provide engineers engaged in the MS&A of aerospace platforms additional configuration flexibility to achieve meaningful results.

CITATION: Wells, J., Hasan, M. and Lucas, C., "Predictive Hold with Error Correction Techniques that Maintain Signal Continuity in Co-Simulation Environments," *SAE Int. J. Aerosp.* 5(2):2012, doi:10.4271/2012-01-2205.

INTRODUCTION

The combination of increasing performance demands, increasing system complexity, and the need for reduced program development schedule and budget costs in the aerospace industry is driving engineers to increasingly rely upon modeling, simulation, and analysis (MS&A) in the platform development cycle. As platform complexity and performance has grown, so too has the fidelity and scale of MS&A that is needed to achieve successful component and subsystem integration at the platform level without adding

significant cost and schedule delays associated with hardware revision required due to undesirable system interactions. In order to maintain reasonable simulation times when examining subsystem interactions at the platform level, it is often necessary to utilize a variety of techniques to accelerate model simulation beyond what can be achieved by simply connecting each of the component models together in a single environment. Such techniques may include average-value modeling [1, 2, 3, 4, 5, 6, 7], singular perturbation [8, 9, 10], multi-rate integration [11, 12, 13], parallel computation [14,

15, 16], FPGA or GPU acceleration [17, 18, 19, 20], amongst many others. One such technique, which offers the advantage that it can generally be applied on top of any of the previously mentioned accelerations and enables integration of models in disparate simulation packages, is co-simulation utilizing technology found in commercially available packages such as PC Krause and Associates, Inc.'s (PCKA's) Distributed Heterogeneous Simulation (DHS) Links or FastSim software [21, 22, 23, 24, 25, 26, 27]. Utilizing this co-simulation approach, significant and potentially super-linear increases in simulation speeds have been shown in many examples [28, 29, 30, 31, 32] with an increase in speed in excess of three orders of magnitude reported in [30].

In co-simulation environments, dynamic models are executed in independent model spaces with each model only exchanging a minimal set of required data typically found at system boundaries and at potentially reduced rates compared to the rate at which the integration routine executes [22] as illustrated in Figure 1. Although this approach enables each model to utilize its own optimized integration routines, event detection algorithms, and data storage approach to achieve faster simulation results, it introduces unique challenges related to the how data exchanged between models is utilized within the model at and between communication events. Although several techniques for such signal estimations have been implemented commercially and described previously in the literature [22, 27], they have not critically studied the effects of prediction error correction at communication events. Typically, such errors are corrected discretely and reduced to zero at the communication event (i.e., they introduce step discontinuities in the signal).

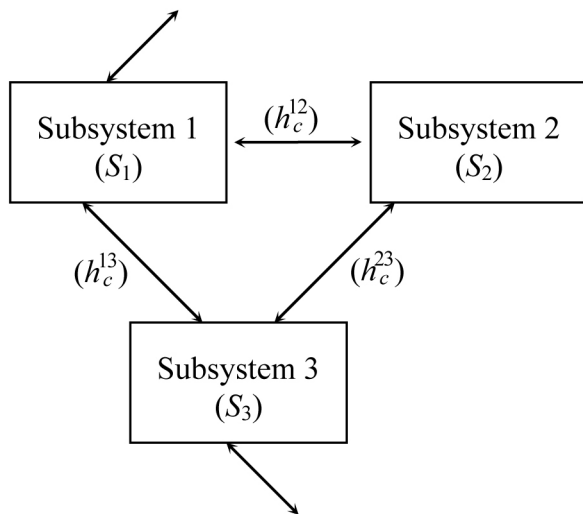


Figure 1. Illustration of the basic architecture of a co-simulation model.

While a discrete error correction approach is conceptually simple, it may not be optimal in certain models due to the injection of step-type excitation signals to the model dynamics. Such lack of optimality can be qualitatively surmised through close inspection of the dynamic model

mathematical descriptions and the numerical integration methods that are utilized to simulate such systems [33]. In such systems, the step function utilized to correct the error introduces significant harmonic content into the estimated signal. Depending upon how the signal is utilized within the receiving subsystem model, there are numerous ways that such harmonic content can reduce simulation speed and accuracy. For instance, if the signal is simply an input to the state model, its impact could be limited to only the impulse excitation of the system. For a well damped system, this could require only a few simulation steps; however, if the system is not well damped, this impulse response could require the simulation to slow down significantly to capture high-frequency oscillatory behavior until the energy decays, at which point, the simulation would be able to proceed normally. In other models, the communicated signal could be integral to the system dynamics directly. In such cases, the discontinuities could have significant impacts on the computation and estimation of the Jacobian of the system model; thereby, resulting in reduced numerical integration accuracy and/or increased numerical effort to determine an appropriate time step for a given error control setting. Finally, such discrete changes in the system can trigger event detection algorithms in the numerical routines or in controllers contained within the model resulting in even further degradation in simulation performance.

In this paper, the prediction and error correction framework in co-simulation environments is described in a general fashion. After the description of a generalized framework, the paper introduces specific alternatives to the traditional error correction approach utilized in co-simulation products, which maintain varying levels of signal continuity. The performance of these alternatives is compared against the traditional approach. Ultimately, these alternative approaches yield more desirable performance under certain conditions and provide engineers engaged in the MS&A of aerospace platforms additional configuration flexibility to achieve meaningful results.

PREDICTION AND ERROR CORRECTION IN CO-SIMULATION COMMUNICATION

As described in the previous section, co-simulation enables the independent configuration of integration routines within subsystem models and the communication of data between these subsystem models. Although not strictly necessary, it is often desirable to configure the communication rate between two subsystems S_i and S_j , h_c^{ij} , to be at a rate that is slower than the rate at which the local integration routine computes the inputs, outputs, and states of the system, h_i^i or h_j^j , respectively. When configured as such, the local subsystem model, S_i , must estimate the values of any data received across the communication interface, y_{ji} , at any time that the integration routine computes the inputs.

outputs, and/or states of the system, t_i^i , based upon previously received values of $y_{ji}^{0...k}$, where $t_{h_c^{ji}}^k$ is the time at which the k^{th} communication event occurred from S_j to S_i and y_{ji}^k is the value of y_{ji} sent from S_j to S_i at time $t_{h_c^{ji}}^k$. Such estimation can be described generally by (1).

$$y_{ji}(t_i^i) = F(y_{ji}^0, \dots, y_{ji}^k, t_i^i) \quad \text{where} \quad t_{h_c^{ji}}^k \leq t_i^i \quad (1)$$

In simulation, the causality of the estimation function, F , is not strictly required and non-causal estimation techniques might often yield improved performance; however, due to practical issues related to non-causal implementations of a co-simulation environment [22], such restriction is generally preferred.

Although F can be arbitrary and may take many particular forms for specific simulations where particular knowledge of the communicated data's form is known a priori, it is common to utilize a polynomial-based predictive hold function due to its relatively simple computational complexity, applicability to signal shapes most commonly transmitted across co-simulation communication channels in physical system models, and its causality. Such polynomial-based predictive hold functions can be expressed generally as

$$y_{ji}(t_i^i) = \sum_{q=0}^n a_{k_{ji}}^q (t_i^i)^q \quad \text{where} \quad t_{h_c^{ji}}^k \leq t_i^i < t_{h_c^{ji}}^{k+1} \quad (2)$$

where n is an integer equal to the order of the polynomial predictive hold function and $a_{k_{ji}}^q$ is the q^{th} order coefficient calculated based upon the last n plus one communications by solving

$$\begin{bmatrix} (t_{h_c^{ji}}^{k-n})^n & (t_{h_c^{ji}}^{k-n})^{n-1} & \dots & (t_{h_c^{ji}}^{k-n})^1 & (t_{h_c^{ji}}^{k-n})^0 \\ (t_{h_c^{ji}}^{k-(n-1)})^n & \dots & \dots & \dots & (t_{h_c^{ji}}^{k-(n-1)})^0 \\ \dots & \dots & \dots & \dots & \dots \\ (t_{h_c^{ji}}^{k-1})^n & \dots & \dots & \dots & (t_{h_c^{ji}}^{k-1})^0 \\ (t_{h_c^{ji}}^k)^n & (t_{h_c^{ji}}^k)^{n-1} & \dots & (t_{h_c^{ji}}^k)^1 & (t_{h_c^{ji}}^k)^0 \end{bmatrix} \begin{bmatrix} a_{k_{ji}}^n \\ a_{k_{ji}}^{n-1} \\ \dots \\ a_{k_{ji}}^1 \\ a_{k_{ji}}^0 \end{bmatrix} = \begin{bmatrix} y_{ji}^{k-n} \\ y_{ji}^{k-(n-1)} \\ \dots \\ y_{ji}^{k-1} \\ y_{ji}^k \end{bmatrix} \quad (3)$$

When described generally, it can be somewhat difficult to follow the various subscripts, coefficients, times, etc.; however, when visualized in an example, the approach is somewhat intuitive. To assist the reader in this visualization, consider the zero, first, and second order hold estimation functions with fixed communication intervals applied to a sinusoidal signal as shown in Figure 2.

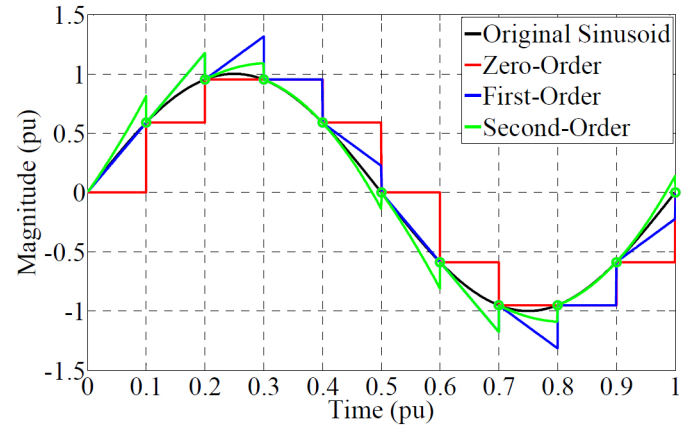


Figure 2. Illustration of the zero-, first-, and second-order hold estimation functions applied to a sinusoidal signal with a sampling rate of ten times the frequency of the sinusoid.

Although the technique described thus far is widely utilized in co-simulation, the estimated waveforms shown in Figure 2 illustrate an interesting and potentially undesirable feature of the estimation scheme: the estimated waveform contains step discontinuities at each communication event when the coefficients are updated. Such discontinuities, as will be discussed in the subsequent sections, can impact numerical integration routines of physical system models in undesirable ways; thereby, resulting in degradation of simulation accuracy and/or increase in simulation communication and computational requirements. Although such discontinuities are readily observed in the estimated signal $y_{ji}(t_i^i)$, they are slightly less obvious but also likely to exist in the derivatives of the signal, which can also impact numerical integration routines though typically to a lesser extent than discontinuities in the signal itself.

To address this discontinuity issue, this paper proposes the conceptual decomposition of the estimation function into two separate components, a prediction function, P , and a prediction error correction function, E , as given by (4).

$$y_{ji}(t_i^i) = P(y_{ji}^0, \dots, y_{ji}^k, t_i^i) + E(y_{ji}^0, \dots, y_{ji}^k, t_i^i) \quad \text{where} \quad t_{h_c^{ji}}^k \leq t_i^i \quad (4)$$

In this new approach, the prediction function P is selected to maximize the accuracy of the prediction similarly to the estimation function F . After the selection of P , the prediction error correction function E is selected to obtain desirable properties in the resulting estimated waveform, presumably while having minimal impact on the overall estimation error. For example, to achieve continuity across communication events in the controlled derivatives, E should satisfy (5)

where the plus and minus signs around $t_{h_c^k}^{ji}$ denote the value of the function in the limit as it approaches $t_{h_c^k}^{ji}$ from either side.

$$\begin{aligned} \frac{d^q}{dt^q} \left(E(y_{ji}^0, \dots, y_{ji}^k, t_l^i) \right) \Big|_{t=t_{h_c^k}^{ji}+} &= \frac{d^q}{dt^q} \left(P(y_{ji}^0, \dots, y_{ji}^k, t_l^i) \right) \Big|_{t=t_{h_c^k}^{ji}+} - \\ \frac{d^q}{dt^q} \left(P(y_{ji}^0, \dots, y_{ji}^k, t_l^i) \right) \Big|_{t=t_{h_c^k}^{ji}-} &- \frac{d^q}{dt^q} \left(E(y_{ji}^0, \dots, y_{ji}^k, t_l^i) \right) \Big|_{t=t_{h_c^k}^{ji}-} \\ \forall \quad q \in \{0, \dots, m\} \end{aligned} \quad (5)$$

Additionally, to minimize the impact on the prediction effort, it may be desirable to have the property that its contribution in the controlled derivative vanishes by the next communication event as described in (6).

$$\begin{aligned} \frac{d^q}{dt^q} \left(E(y_{ji}^0, \dots, y_{ji}^k, t_l^i) \right) \Big|_{t=t_{h_c^{k+1}}^{ji}} &= 0 \\ \forall \quad q \in \{0, \dots, m\} \end{aligned} \quad (6)$$

For example, if P is taken to be the polynomial predictive hold function previously described, a potential selection for the function E is

$$\begin{aligned} E(t_l^i) &= \sum_{q=0}^{2m+1} b_{kji}^q (t_l^i)^q \quad \text{where} \quad t_{h_c^k}^{ji} \leq t_l^i < t_{h_c^{k+1}}^{ji} \\ &\quad \text{and} \quad m \leq n \end{aligned} \quad (7)$$

where n is the polynomial order of P , m is the order of derivative that is desired to be continuous (i.e., zero will ensure continuity in the signal, one in the first derivative, two in the second derivative, etc.) and b_{kji}^q is the q^{th} order coefficient calculated by solving

$$\begin{bmatrix} \mathbf{A}_k \\ \mathbf{A}_{k+1} \end{bmatrix} [\mathbf{b}_k] = \begin{bmatrix} \mathbf{M}_k [\mathbf{a}_k - \mathbf{a}_{k-1}] \\ 0 \end{bmatrix} \quad (8)$$

at the k^{th} communication event where $\mathbf{A}_k$ and $\mathbf{M}_k$ relate to the equations utilized to enforce continuity at the k^{th} communication event as given by (5) and $\mathbf{A}_{k+1}$ relates to the equations that enforce the nullification of the contribution of the prediction error correction function at the $(k+1)^{\text{th}}$ communication event as given by (6), with each as defined in (9), (10), (11), respectively.

$$\begin{aligned} \mathbf{A}_k(p, q) &= \begin{cases} \left(\frac{((2m+2)-q)!}{((p-(m+1))+((2m+2)-q))!} \right) \left(t_{h_c^k}^{ji} \right)^{(p-(m+1))+((2m+2)-q)} \\ 0 \end{cases} \\ \forall \quad &\begin{cases} p \in \{1, \dots, m+1\} \\ q \in \{1, \dots, (2m+2)+(p-(m+1))\} \\ \text{otherwise} \end{cases} \end{aligned} \quad (9)$$

$$\begin{aligned} \mathbf{M}_k(p, q) &= \begin{cases} \left(\frac{((n+1)-q)!}{((p-(m+1))+((n+1)-q))!} \right) \left(t_{h_c^k}^{ji} \right)^{(p-(m+1))+((n+1)-q)} \\ 0 \end{cases} \\ \forall \quad &\begin{cases} p \in \{1, \dots, m+1\} \\ q \in \{1, \dots, (n+1)+(p-(m+1))\} \\ \text{otherwise} \end{cases} \end{aligned} \quad (10)$$

$$\begin{aligned} \mathbf{A}_{k+1}(p, q) &= \begin{cases} \left(\frac{((2m+2)-q)!}{((p-(m+1))+((2m+2)-q))!} \right) \left(t_{h_c^{k+1}}^{ji} \right)^{(p-(m+1))+((2m+2)-q)} \\ 0 \end{cases} \\ \forall \quad &\begin{cases} p \in \{1, \dots, m+1\} \\ q \in \{1, \dots, (2m+2)+(p-(m+1))\} \\ \text{otherwise} \end{cases} \end{aligned} \quad (11)$$

Although (9), (10), (11) are somewhat complicated in notation, they can be simply described as stating that the

bottom row is a vector with the time $t_{h_c^*}^{ji}$, where $*$ is either k or $k+1$, raised to each power from $2m+1$ to zero for $\mathbf{A}_*$ (or $n+1$ to zero for $\mathbf{M}_k$). Each row moving up takes the derivative of the row below with respect to time and then shifts the resulting vector to the left by one index resulting in a pseudo lower-triangular matrix. The vector $\mathbf{b}_k$ is a column vector (of length $2m+2$) of the coefficients to be calculated at the k^{th} communication event in descending order with the highest order coefficient at the top of the vector while $\mathbf{a}_k$ and $\mathbf{a}_{k-1}$ are similar column vectors (of length $n+1$) for the predictive hold function at the k^{th} and $(k-1)^{\text{th}}$ communication event, respectively. Note that MATLAB™-style one-based indexing has been assumed throughout this paper.